Spring 5-1-2003

Analysis and Design of Sales Data Warehouse

Muhammad Saleem
Dakota State University

DAKOTA STATE UNIVERSITY

MSIS

PROJECT APPROVAL FORM

Student Name: Muhammad Saleem

Expected Graduation Date: May 10 , 2003

Master's Project Title: Analysis and Design of Sales Data Warehouse

Date Project Plan Approved: _____

Date Project Coordinator Notified and Grade Submitted: _____

Approvals/Signatures:

Student: _____ Date: 9-10-03

Faculty supervisor: _____ Date: 9-12-03

Committee member: _____ Date: 9/12/03

Committee member: _____ Date: 9/12/03

Committee member: _____ Date: 6/8/03

Copies to:

Original attached to Written Report

Copies to: Advisor, Graduate Coordinator and Student

Solo Cup
Analysis and Design of Sales Data Warehouse

BY
Muhammad Saleem

A report submitted in partial fulfillment of the requirements for the degree of

Master of Science in Information Systems
Dakota State University

2003

Committee Members

Project Supervisor:	Dr. Ronghua Shan, DSU College of BIS
Committee member:	Dr. Terry Dennis, DSU College of BIS
Committee member:	Assistant Prof. Stephen Krebsbach, DSU College of BIS
Committee member:	Sartaj H. Iram, Research Assistant, UOI, Urbana

Acknowledgements

I would like to thank:

Dr. Shan, my Project Supervisor, for his continuous help, guidance and patience, which helped me in completing my project report swiftly.

Dr. Terry Dennis, my Advisor, who taught us how to handle a project in real life problems.

Prof. Stephen, whose teachings leaded me to select a data warehouse project for my final IS Project.

Mr. Iram, who was my "Proctor" for several of my courses and now he has committed himself to be part of external Project Committee Member.

Laurie Dennis, for her continuous guidance to us for completing our course work and other curriculum and non-curriculum activities.

My wife, who always helped me in managing my time to successfully complete all of my MSIS courses.

Abstract

I have been working in the field of Information Technology since my graduation in Computer Sciences from Gomal University of Pakistan in 1992. I have been working as an Oracle Database Administrator since 1995 and databases is one of my favorite subjects and that is why I chose Data Management concentration in MSIS program. After attending INFS 684, INFS 786 and INFS 787, I decided to work on data warehouse project as part of MSIS partial requirement. I was already aware of the data warehouse importance as being the part of database management. In this competing business environment the information technology is playing a major role and data warehouse or decision support systems have the leading edge on business competitors. However, this is also true that several companies are still struggling to build the data warehouses and decision support systems. In my whole experience as an Oracle DBA, I saw very few companies who developed some of the state of the art decision support systems. Some companies are just dumping their online transaction systems in separate databases as it is. When such databases grow, the reports run forever because of not having proper data warehouse design which could have data spread across dimensions and some de-normalization according to reports requirements. In this project, I have tried to design a data warehouse for Solocup, which will address all of the issues discussed above.

Table of contents

List of Appendices

List of Figures

1. Introduction

Solocup Company was established in 1936 as a start-up company entitled the Paper Container Manufacturing Company in Chicago. Leo J. Hulseman was its founder and president, producing a line of disposable cups.

In the 30's and the 40's this company thrived in the disposable business, making cold drink cups. They established themselves at the leading edge of disposable technology, with the creation of the PC cone cup. In 1946, a major change occurred. The popular PC cone cup was renamed the Solo Cup. Shortly thereafter the company name was also changed to its current name - Solo® Cup Company.

The use of disposable cups in the 1950's became popular and disposable cups were being used more than ever. Company expanded its business with every passing year and introduced another new technology - 2 pieces waxed cold cups - used in many fast food restaurants today. Now Solocup Company has a large number of disposable paper and plastic products and is the biggest supplier to some of Giants retail stores like Wal-Mart and K-Mart.

Today, Solo Cup Company owns and operates 14 manufacturing plants and 12 warehouses/distribution centers. These production facilities supply business owners and consumers with a wide variety of disposable products ranging in composition from plastic, to foam, to paper. Commitment to quality products at affordable prices is the hallmark of Solo Cup Company.

Company has several departments, which are fully automated with state of the art computer technology. There is a separate IS (Information Systems) department

which provides support for company's day to day computing requirements along with maintenance of the existing infrastructure. There are several databases supported by IS department like Order Management, Rebates.

2. Statement of Problem

Solocup Company started its business with few manufacturing units and since that company is continuously expanding its business. Closure of the competitor's chain provided a great opportunity for the company to acquire more manufacturing units and clients and now company has presence in several states.

There are 12 warehouses where the manufactured and finished products are kept before they are sold to the whole sellers and retailers. Each location has transactional system, which keeps track of products in and out from the warehouse. There is a central warehouse database, which is updated directly from several warehouses systems using a virtual private network. The Order Management System keeps track of customer orders. Rebate database keeps tack of several promotions and rebates offered to customers. The Head office runs reports against Warehouses, Order Management and Rebate systems to figure out the sales to the whole sellers, retailers, restaurants and several fast food franchises.

The challenge faced by Information Services department is to produce strategic reports, which could span on months and years of data. Also sometimes it is almost difficult to produce a report that is asked by the management due to limited data and resources. The sales are difficult to track region wise with the existing system and exact inventory know how at each location is a nightmare. Management is always interested in getting the historical data for strategic decisions like over all sales by region, customer and product, know how about markets and employees in sales department. Moreover operational databases have grown to such a stage where

queries take long time to execute and operational systems cannot afford to be taxed heavily for the strategic reports.

3. Project Objectives

Jim Kearns, CIO, has recently joined the Solocup and is very ambitious to streamline several of the processes and from very first day he took some of the bold steps; one of them was to develop a Sales data warehouse for the company because operational system were being heavily taxed by long running queries for strategic decisions. Also due to operational systems limit some of the historical data may not be available for reporting purposes. I was working as a Senior Oracle DBA for this company for past year and half and I was aware of several of the business processes and operational system and I knew how the data warehouse would be developed which new CIO was thinking. I offered myself to CIO to accept this challenge and also told him about my recent courses in data management from Dakota State University. He told me to move forward with the plan to design and develop a Sales Data Warehouse.

4. Project Planning

4.1 Work Break Down Structure

I initiated my MSIS Project Idea on September 04, 2002 and decided to work on a data warehouse project. Since I had been working in Solocup Company for more than a year, it was appropriate to use my knowledge and skills, learned in MSIS program, on my existing job. To meet my project objectives I started developing a work break down structure (Appendix I), which was a baseline to move through the project.

4.2 Gantt Chart

After developing a work break down structure, I started making detailed tasks list and identified the dependencies and priorities of the tasks. To meet project objectives, I distributed tasks in four groups:

- Planning
- Analysis
- Design
- Implementation

The estimated time to complete the project was 161 days including 89 days for planning, 36 days for analysis, 25 days for design and 11 days for implementation. Complete project plan is attached as appendix J.

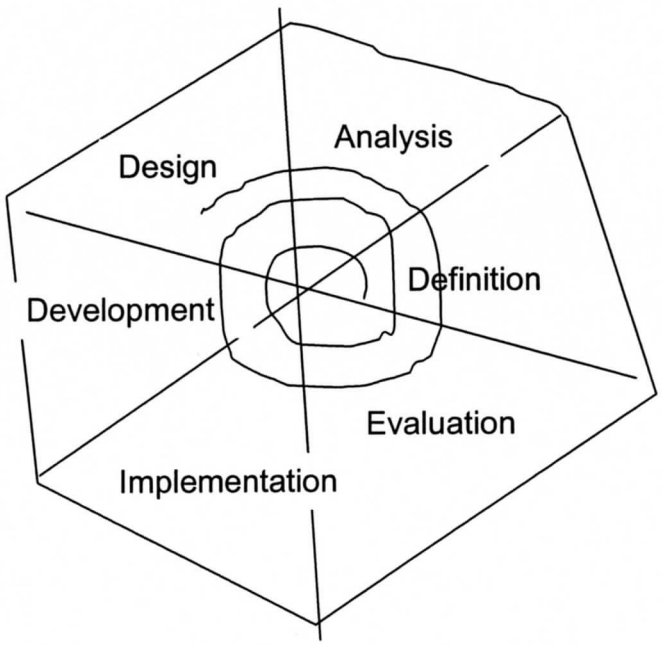
5. Analysis

5.1 The Development Spiral

A data warehouse development methodology must have three basic characteristics. First, it must be flexible. The data warehouse construction team should be able to begin the development process without a detailed definition of every screen and data element in the system. Second, the team members should be able to construct a system that will meet the diverse and often competing needs of the user community. Finally, the development process must be a quick and right.

To meet the demands of the data warehouse we are using an approach, the Development Spirals as shown in figure below. The spiral methodology begins modestly and expands outward. With each pass through the spiral, however, new functionality is added, expanding the spiral outward. At the end of each pass through the spiral there is an evaluation phase.

(Figure 4.1)



5.2 Interviews

I have chosen many people from Solocup to be interviewed and to know what each and everybody wants and what are their expectation levels. Please see the appendix "A" for a template letter, which I used to invite interviewees. I'm glad Jim Kearns, who recently joined Solocup as CIO has great personal interests in developing a data warehouse. In his first address to the IT department people, he made clear about his adventures and he set aside goals for the IT department for over the next two years. Among these goals it was the development of a new data warehouse, which would help company to make strategic decisions in the long run.

I interviewed several people (see Appendix B) working in different capacities according to my initial plan. I could not interview anybody in financial department, as they were extremely busy in the year-end and then migrating existing financial system to the new hardware and software releases.

5.3 Data Warehouse Readiness Litmus Test

In the analysis phase it is very important to determine the environment under which the data warehouse will be developed. Appendix C lists all the main factors, which I tested to know the data warehouse readiness of the Solocup. Following is a brief discussion about several data warehouse readiness factors.

5.3.1 Strong Business Management Sponsor

The new CIO, Jim Kearns looks very keen in develop a comprehensive data warehouse. When I told him about my recent courses related to data management, he asked me several questions about the data warehouse requirements. Solocup

already has a state of the art data center and enough hardware/software resources are available. But even then he bought few more powerful data servers (HP 8400 with 8 processors and 16g of memory). Soon after I completed interviews, he asked Jim Lail to hire a new Data Warehouse Project Manager. Due to Jim Kearns personal interests I think this project will get a very strong sponsor eventually.

5.3.2 Compelling Business Motivation

In a very first address Jim Kearns sold his ideas to the business owners and users and told about his experiences in past shops. He also promised to setup a new vision for the company within next three months. After three months he came up with an integrated solution and he decided to use Middle Ware to integrate all of the business interfaces along with the idea of developing a comprehensive data warehouse. There were many complaints from the managers and business users for lack of data and resources availability for decision-making. So I think there is a compelling business motivation present in developing a data warehouse for the Solocup Company.

5.3.3 IS/Business Partnership

Solocup is using IT resources since 80's and there is a very high level of confidence in using computer resources. There is a separate IS department which provides and facilitates other departments. The new IS management has recently added an Help-Desk section of IS department to provide better and quick services to business users. Business users rely heavily on IS department when it comes to the information. So I think there is a strong IS and Business partnership in Solocup.

5.3.4 Current Analytical Culture

As I mentioned before, business users heavily rely on IS department for data and information. There are several database applications, which are being used by the business users. Data is directly keyed in databases using front-end applications and reports are frequently requested from the IS department. Information is shared openly throughout the Solocup Company and overall computer know how of business users is very high.

5.3.5 Feasibility

Solocup has robust technical structure in place and there will be minimum additional requirements from hardware and software standpoint. Solocup also has experienced staff in IS department and planning to hire more experienced people with a solid data warehouse experience. Online systems are very stable and running for a long time, which will provide quality data for data warehouse.

5.4 Analysis Model

Most of the sales reports are generated in the head office from Sales History database. The current Sales History database is populated from online Sales and Rebate databases. The Sales database has all the information for customers and orders. This database also keeps track of shipment and delivery information, which is updated from twelve different Warehouses, which send data in text files each day though electronically. Accounting department also get reports of shipment and delivery through snapshots of data from Sales Database at the end of everyday. They send invoices to customers based on the information, which is extracted from

financial database. Brokers also have some restricted access to Rebates and Sales databases.

As we are focusing on design of the data warehouse for the company we will be mostly concentrating on transactional systems like Sales and Rebates databases in the head office, which will provide all the basic aggregates of data for the data warehouse.

Following objects has been identified from the discussions at different levels:

Head Office

District Sales Offices

Warehouses

Brokers

Retail Sellers

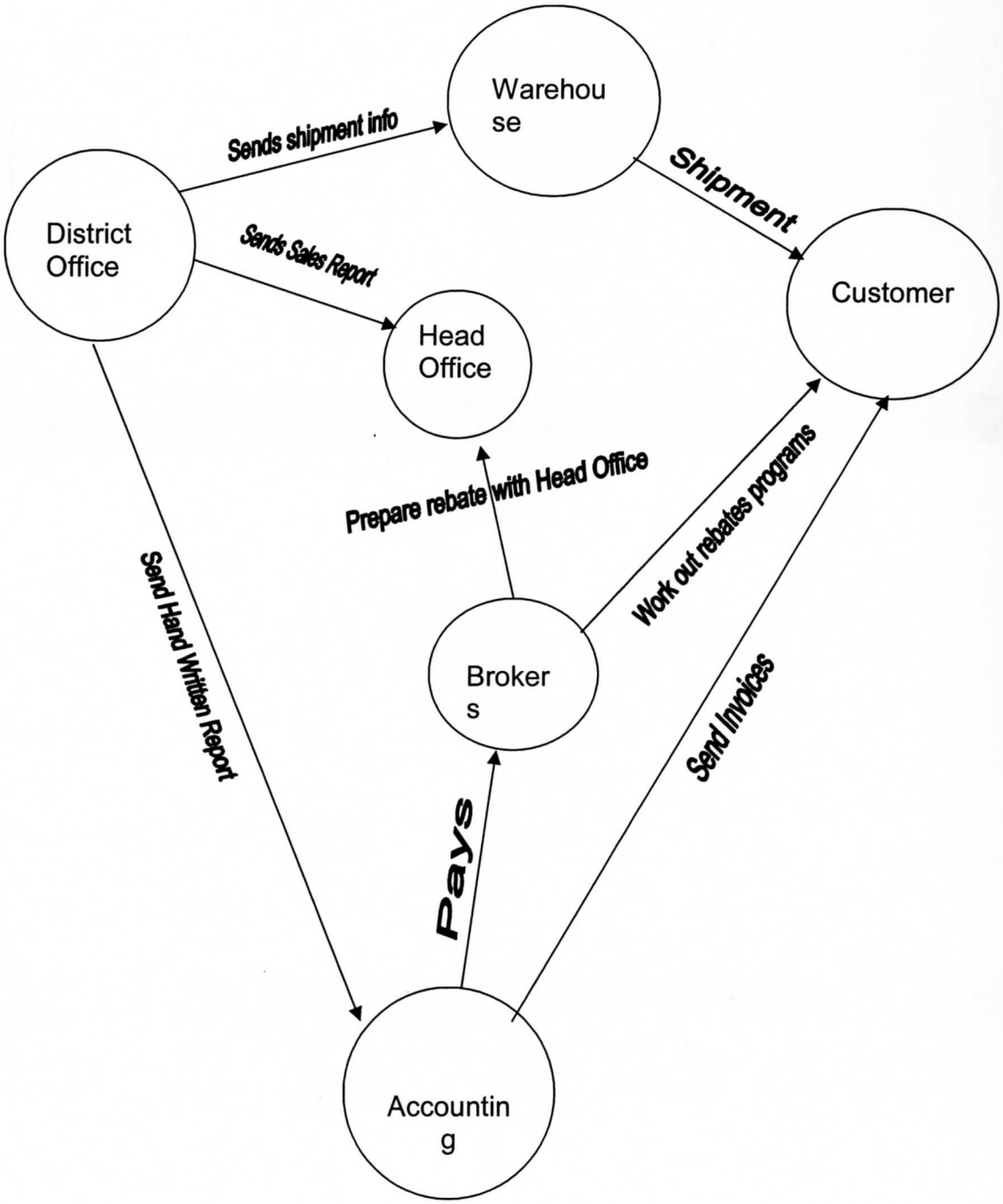
Restaurants

Accounting

Customers

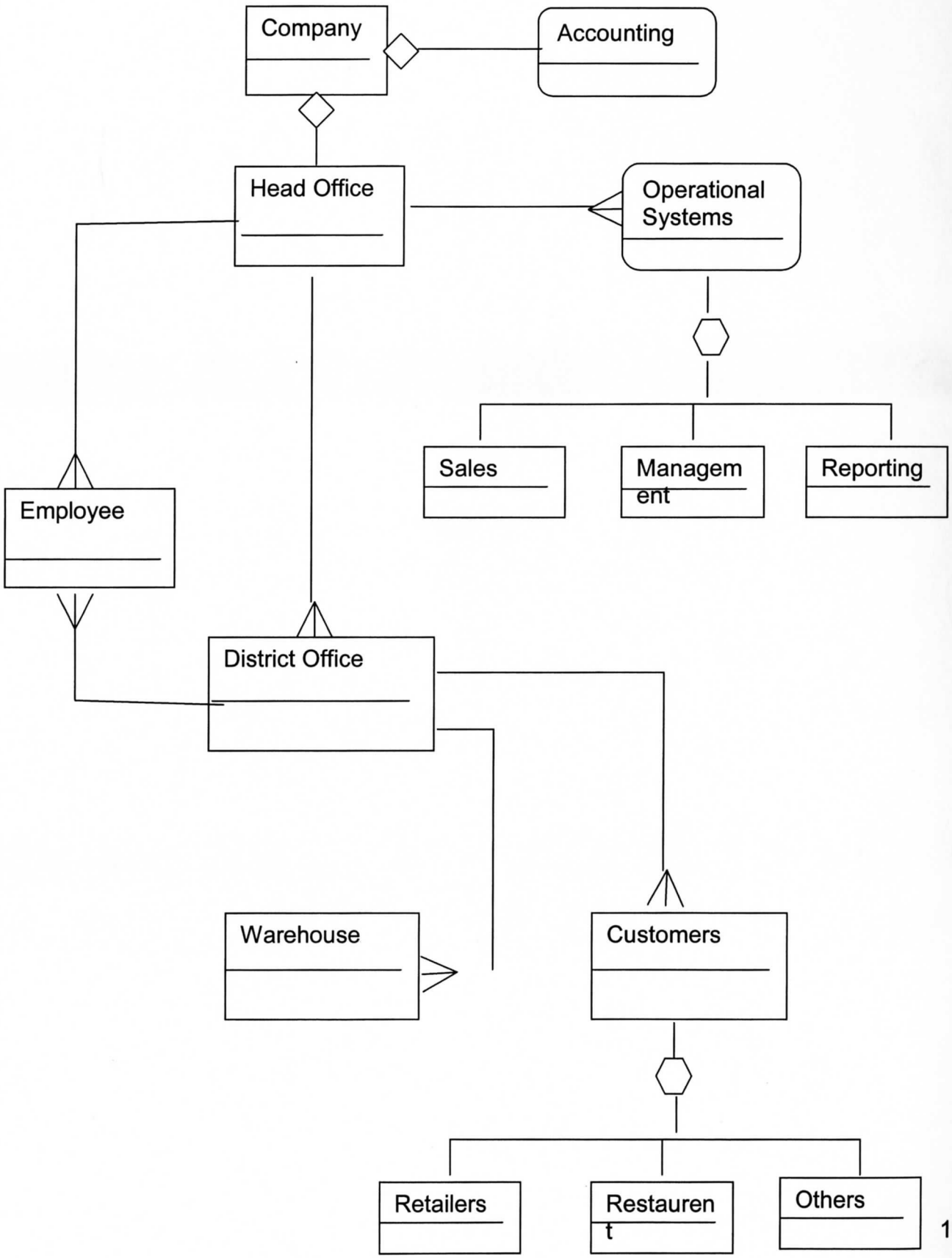
Analysis Model

(Figure 4.2)


Warehou
se
Sends shipment info
Shipment
District
Office
Sends Sales Report
Customer
Head
Office
Prepare rebate with Head Office
Work out rebates programs
Send Hand Written Report
Broker
s
Pays
Send Invoices
Accountin
g

5.5 Object Relational Model

(Figure 4.3)



5.6 Analysis of Hardware and Software Requirements

Solocup has a state of the art data center at a centralized location, and most of the infrastructure is already there. Here is a brief list of hardware and software

4000GB DataTek Storage Solutions

K-Class, RP2405, RP5400 and RP8400 HP Servers

High Speed Storage Tek Tape library

Operating System - HP-UX 11I on all servers

Enterprise Wide database solutions – Oracle Database 8.1.7 and 9.2.0.2

Giga Bit backbone Ethernet Switch

Most employees are equipped with the latest PCs having latest software

So based on above infrastructure, we may not need any new hardware and software as some of the existing servers are very powerful and they can be used for installing this new Sales Data Warehouse without any issues.

6. Design

6.1 Logical Data Model

Based on information in analysis phase, we are now moving to the design of data warehouse objects and schema. When we talk about the way we want to look at data, we usually want to see some sort of aggregated data. These data are called measures. These measures are numeric values that are measurable and additive. For example, our sales dollars are a perfect measure. Every order that comes in generates a certain sales volume measured in some currency. If we sell twenty products in one day, each for five dollars, we generate 100 dollars in total sales. Therefore, sales dollars is one measure we may want to track. We may also want to know how many customers we had that day. Did we have five customers buying an average of four products each, or did we have just one customer buying twenty products? Sales dollars and customer counts are two measures we will want to track.

Just tracking measures isn't enough, however. We need to look at our measures using those "by" conditions. These "by" conditions are called dimensions. When we say we want to know our sales dollars, we almost always mean by day, or by quarter, or by year. There is almost always a time dimension on anything we ask for. We may also want to know sales by category or by product. These by conditions will map into dimensions: there is almost always a time dimension, and product and geographic dimensions are very common as well.

Therefore, in designing a star schema, our first order of business is usually to determine what we want to see (our measures) and how we want to see it (our dimensions).

Dimension tables answer the "why" portion of our question: how do we want to slice the data? For example, we almost always want to view data by time. We often don't care what the grand total for all data happens to be. If our data happen to start on June 14, 1989, do we really care how much our sales have been since that date, or do we really care how one year compares to other years? Comparing one year to a previous year is a form of trend analysis and one of the most common things we do with data in a star schema.

We may also have a location dimension. This allows us to compare the sales in one region to those in another. We may see that sales are weaker in one region than any other region. This may indicate the presence of a new competitor in that area, or a lack of advertising, or some other factor that bears investigation.

When we start building dimension tables, there are a few rules to keep in mind. First, all dimension tables should have a single-field primary key. This key is often just an identity column, consisting of an automatically incrementing number. The value of the primary key is meaningless; our information is stored in the other fields. These other fields contain the full descriptions of what we are after. For example, if we have a Product dimension (which is common) we have fields in it that contain the description, the category name, the sub-category name, etc. These fields do not contain codes that link us to other tables. Because the fields are the full descriptions, the dimension tables are often fat; they contain many large fields.

Dimension tables are often short, however. We may have many products, but even so, the dimension table cannot compare in size to a normal fact table. For example, even if we have 30,000 products in our product table, we may track sales for these products each day for several years. Assuming we actually only sell 3,000 products in any given day, if we track these sales each day for ten years, we end up with this equation: 3,000 products sold X 365 day/year * 10 years equals almost 11,000,000 records! Therefore, in relative terms, a dimension table with 30,000 records will be short compared to the fact table.

6.2 Star Schemas

Based on the interviews and analysis model, I moved forward with design of following star schemas.

6.2.1 Employee-Product Schema

This schema will address the reporting requirements to determine which employee made how much Dollar sale for each product in a day, week, month or year. There will be three dimensions in this star schema EMPLOYEE, TIME and PRODUCT and a fact table. The EMPLOYEE dimension will be populated from financial database because employee's data is maintained centrally in HR database. The PRODUCT dimension will be populated from Order Management (Sales Database). EMPLOYEE and PRODUCT dimension will be required to refresh from HR and Order Management databases respectively whenever new employees or products are added.

6.2.2 Customer-Product Schema

This schema will address the reporting requirements to determine how many products are sold to each customer in a day, week, month or year. There will be three dimensions in this star schema CUSTOMER, TIME and PRODUCT and a fact table. The CUSTOMER and PRODUCT dimension will be populated from Order Management (Sales Database). CUSTOMER and PRODUCT dimension will be required to refresh from the Order Management database whenever new customer or products are added.

6.2.3 District-Product Schema

This schema will address the reporting requirements to determine how many products the each sales district sells in a day, week, month or year. There will be three dimensions in this star schema DISTRICT, TIME and PRODUCT and a fact table. The DISTRICT and PRODUCT dimension will be populated from Order Management (Sales Database). DISTRICT and PRODUCT dimension will be required to refresh from the Order Management database whenever new district or products are added.

6.2.4 Rebate-Product Schema

This schema will address the reporting requirements to determine how much rebate is paid for each product in a day, week, month or year. There will be three dimensions in this star schema REBATE, TIME and PRODUCT and a fact table. The REBATE dimension will be populated from rebate database and The PRODUCT dimension will be populated from Order Management (Sales Database). The REBATE and PRODUCT dimension will be required to refresh from REBATE and

Order Management databases respectively whenever new rebates are offered to customers.

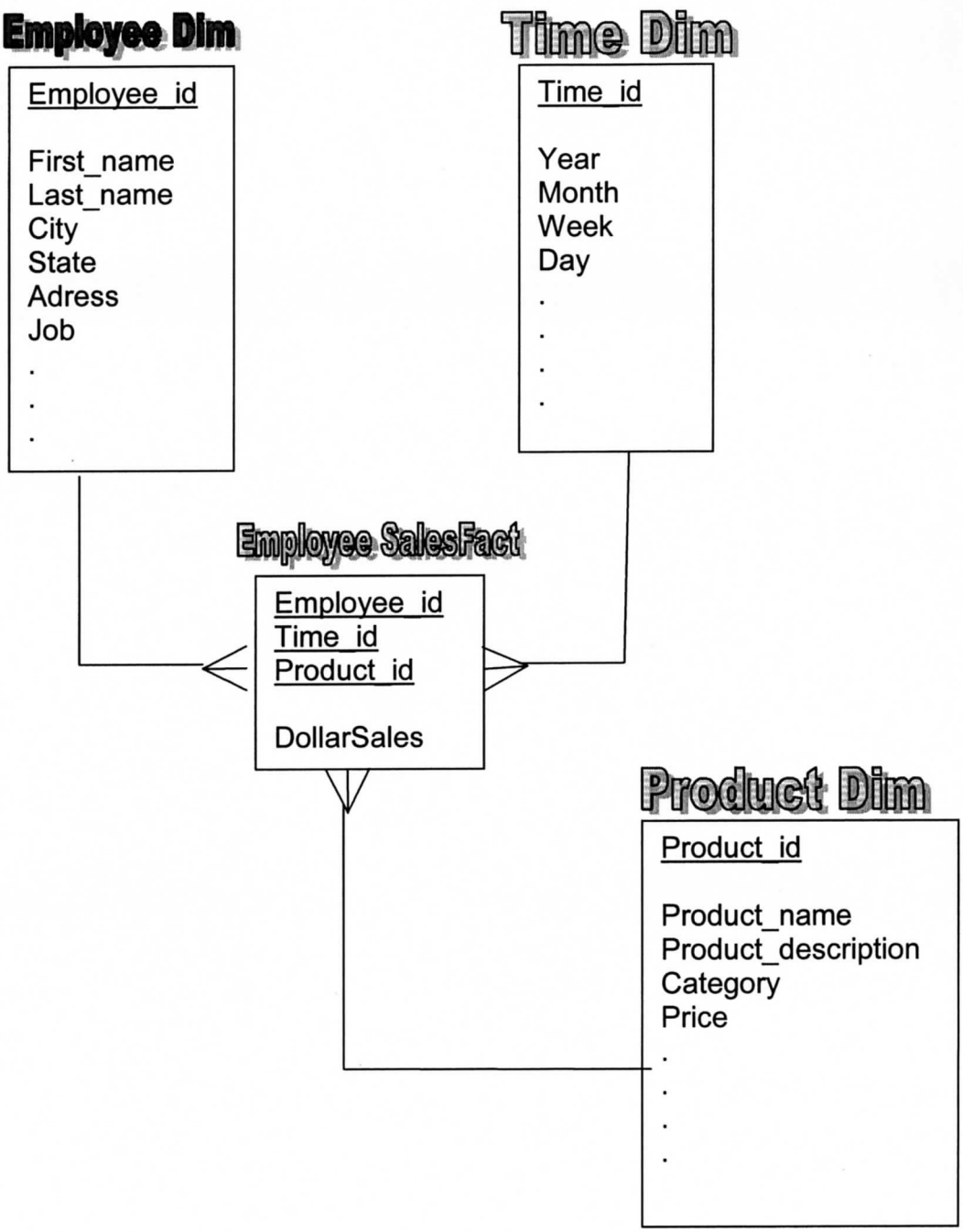
6.2.5 Product Schema

This schema will address the reporting requirements to determine how much each product is sold in a day, week, month or year. There will be two dimensions in this star schema PRODUCT and TIME. The PRODUCT dimension will be populated from Order Management (Sales Database). The PRODUCT dimension will be required to refresh from Order Management databases whenever new products are added.

6.3 Star Schemas ERD Diagrams

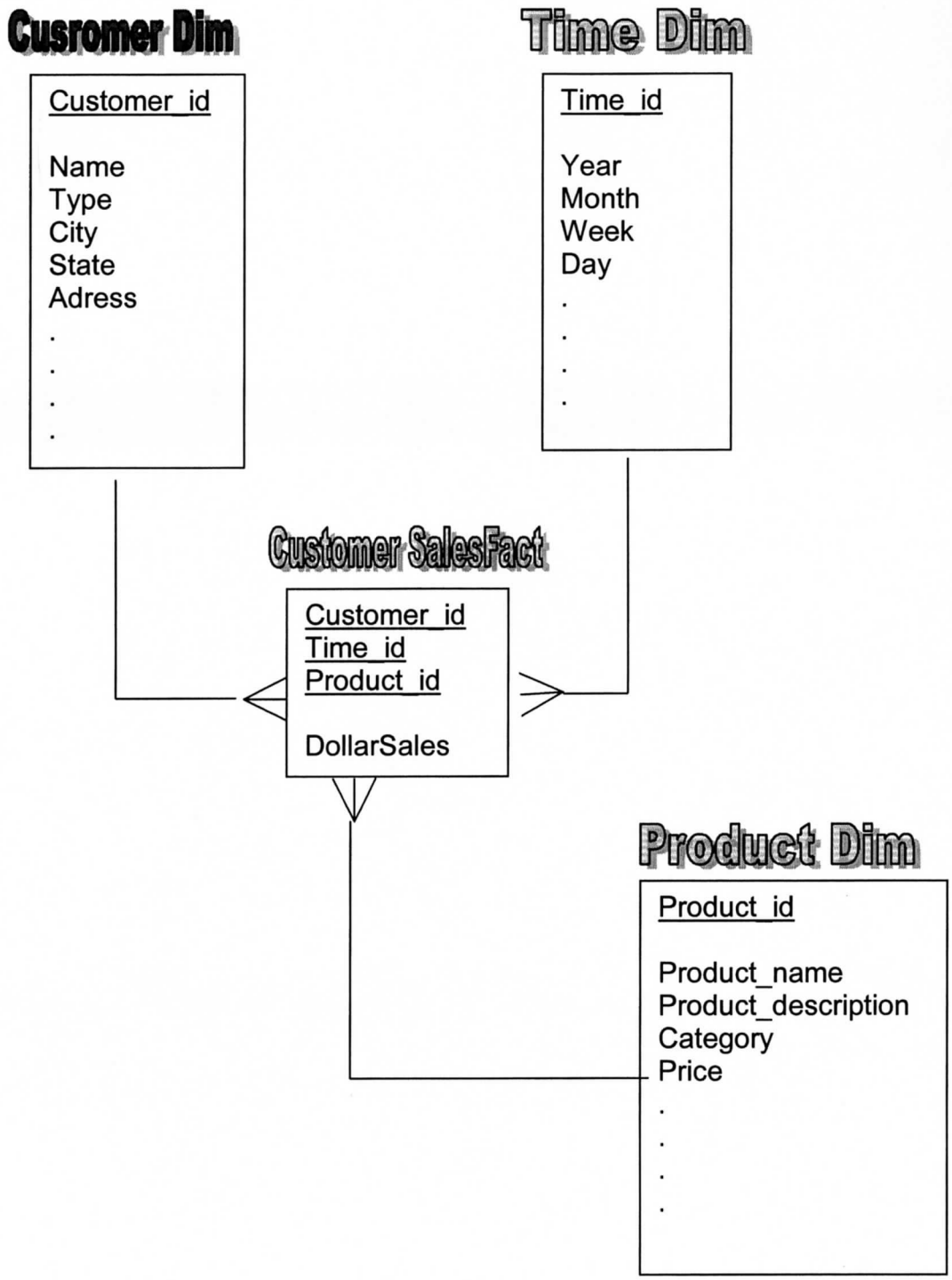
6.3.1 Employee-Product Sales Dimensional Model

(Figure 5.1)



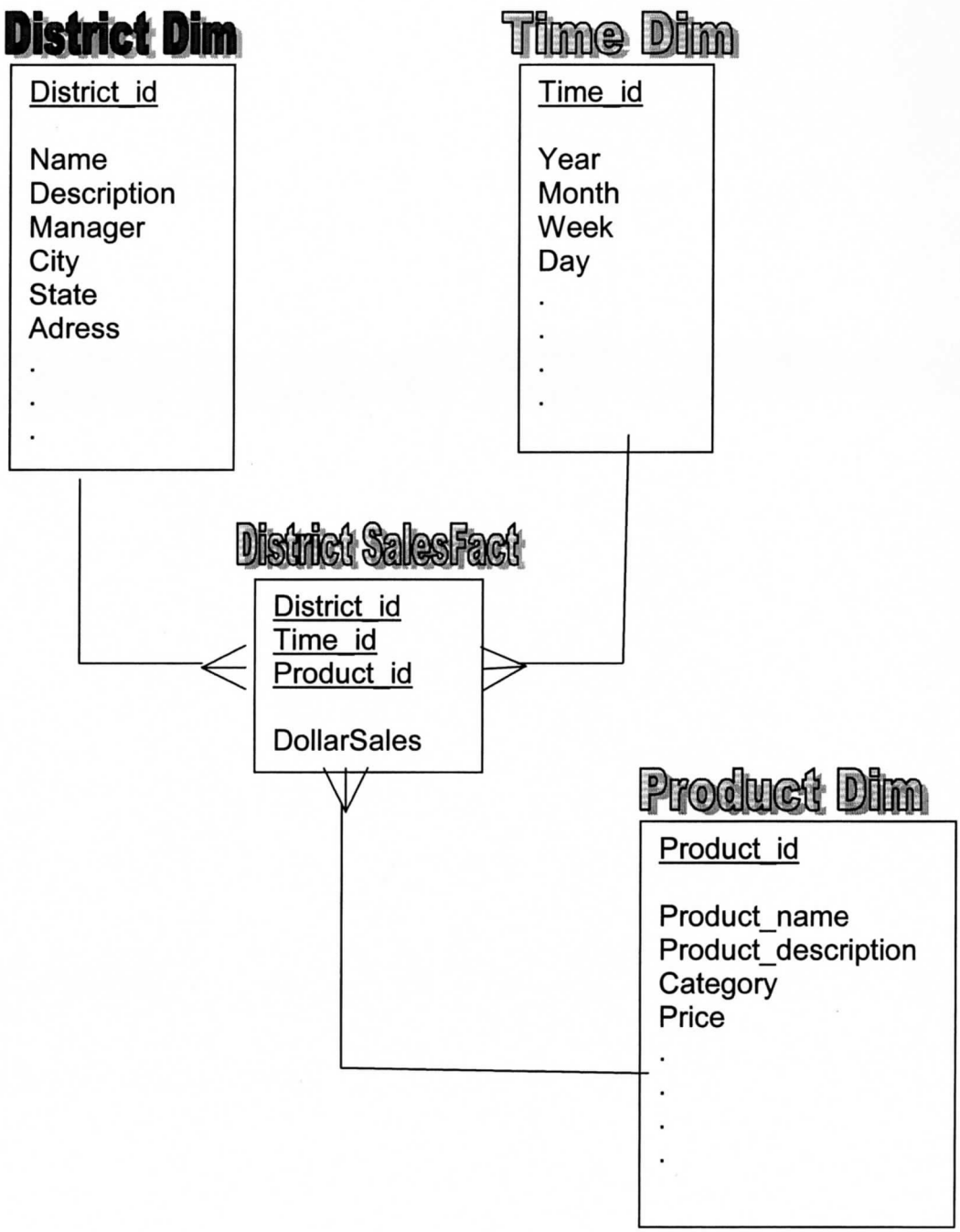
6.3.2 Customer-Product Sales Dimensional Model

(Figure 5.2)



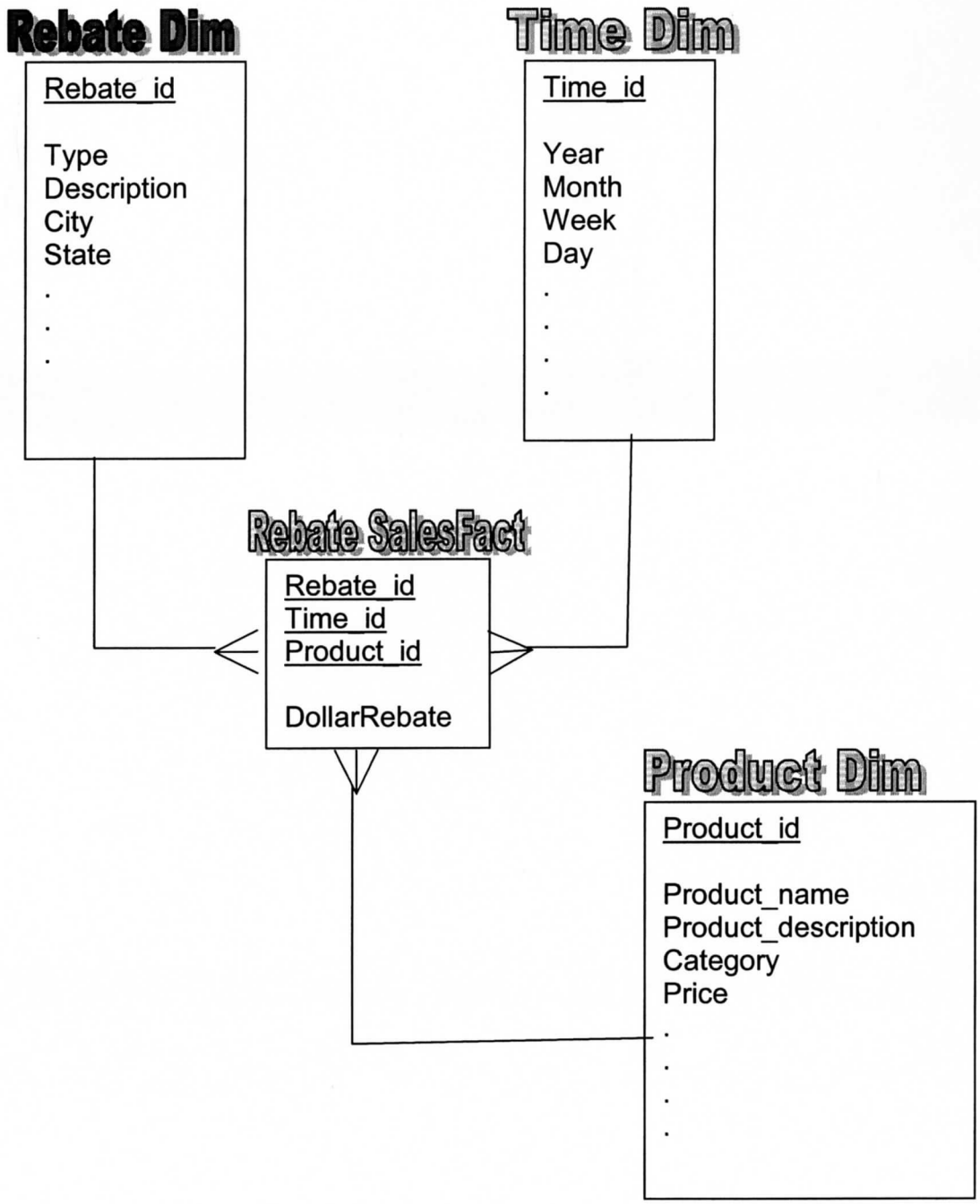
6.3.3 Districrt-Prodcut Sales Dimensional Model

(Figure 5.3)



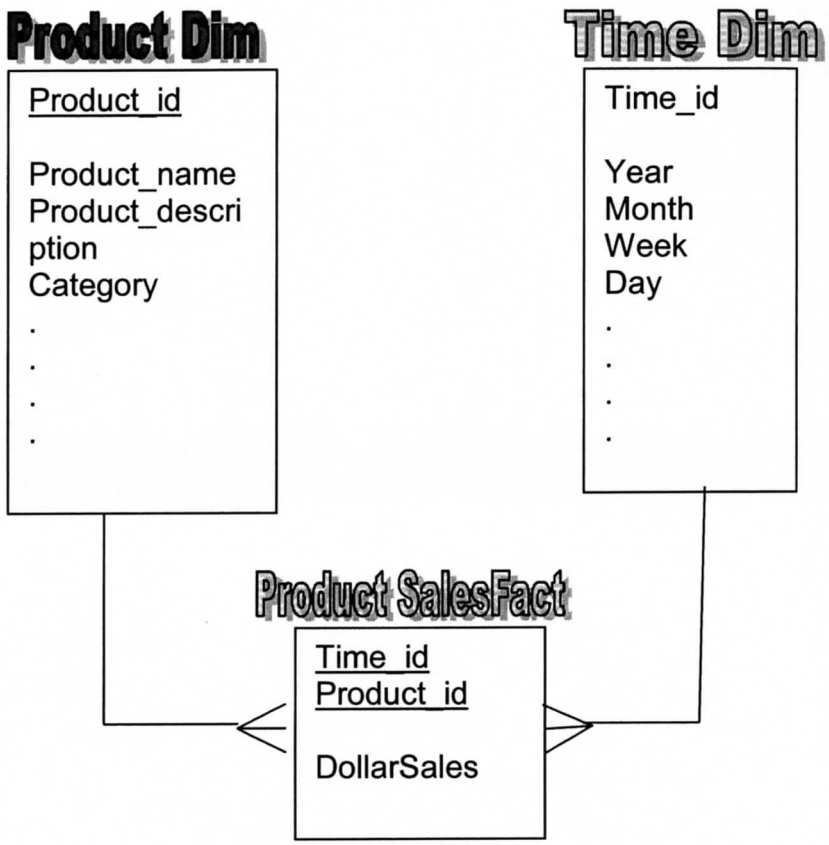
6.3.4 Rebate-Product Sales Dimensional Model

(Figure 5.4)



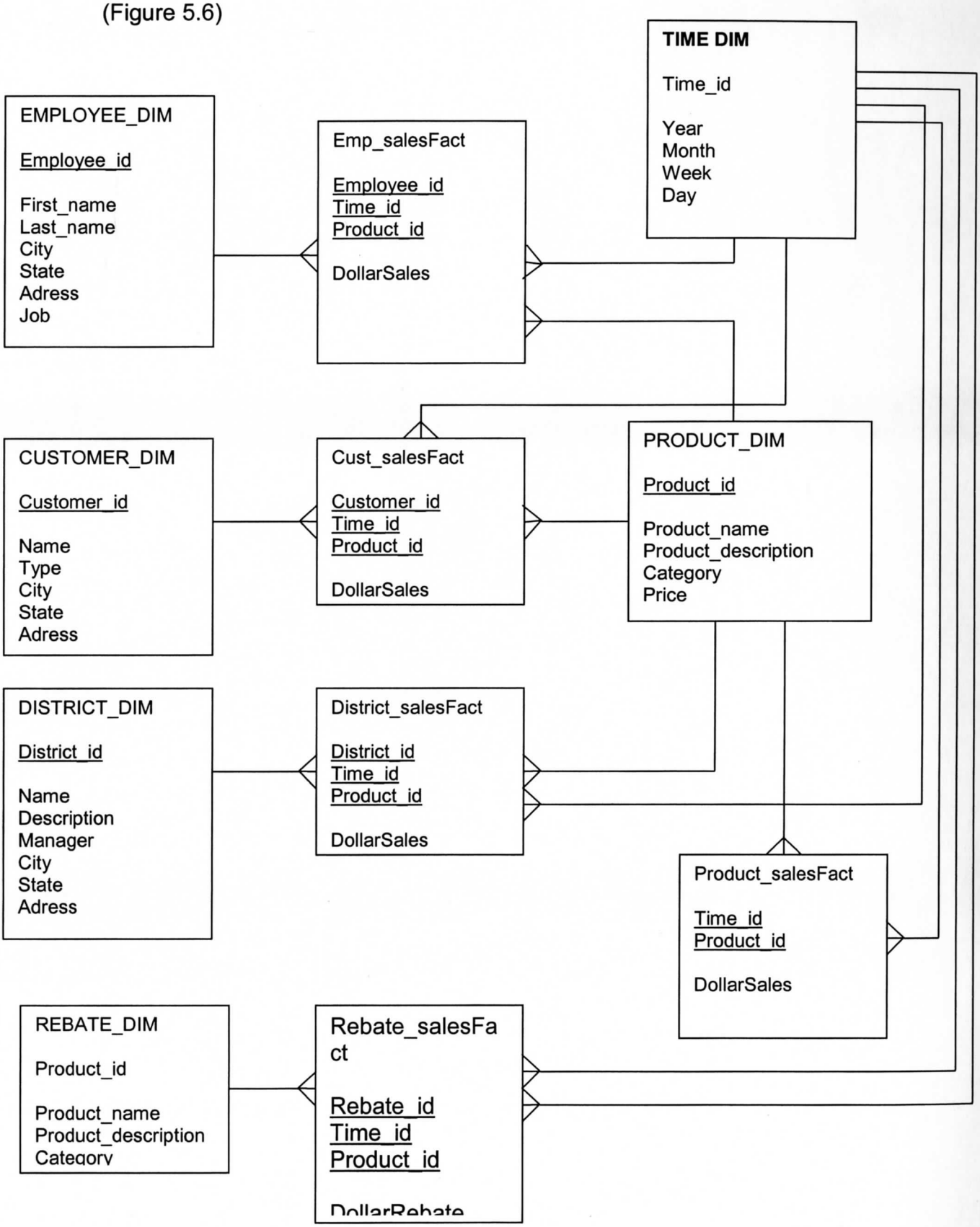
6.3.5 Product Sales Dimensional Model

(Figure 5.5)



6.4 Snow-Flake Schema

(Figure 5.6)



6.5 Data Staging Model

Data extraction from operational systems is one of the very crucial part of the data warehouse. In our Sales Data Warehouse we will extract data from Order Management System, Rebate Management System and Financial System using Oracle snapshots with fast refresh option. Following is a list of tables that we will replicate from each of above systems.

Order Management System: We will replicate PRODUCT, ORDER, CUSTOMER, SALES_DETAIL. ORDER and SALES_DETAIL table will be the only tables, which will have significant amount of data every day to be replicated.

REBATE Management System: REBATE, REBATE_DETAIL and REBATE_CATEGORY tables will be replicated from this system. Since rebates are run few times in a year, so there will be no significant amount of data to be replicated everyday.

Financials Management System: Since new customers are always added through financial systems also, so we will only replicate CUSTOMER table from Financial Management Systems at this time. Later, in future, we can bring more historical data into data warehouse as per Finance Department requirements.

Data Verification: Data verification will be done in staging area. Staging area will have a complete record of last data warehouse refresh and according to the previous information, timestamps will be added to the data.

Data Verification Logs: Staging Management System will keep track of all of the data and its integrity and will generate log files, which will be reviewed, by system management team, on a regular basis.

Data Loading: After data is verified and logged, it will be loaded in Sales Data Warehouse. Automated scripts which development team will write for us will control data verification, logging and all other steps in data staging area as shown in figure on next page.

6.6 Data Validation Checklist

6.6.1 Search for Common Data Problems

Inconsistent or incorrect use of codes and special characters

Single field used for unofficial or undocumented purposes

Overloaded codes

Evolving data

Missing, incorrect, or duplicate values

6.6.2 Ensure Proper Name and Address Handling

Name and address split into individual components

Individual components cleaned and corrected

All appropriate components completed

Duplicates eliminated

Cleaned data fed back into source systems

Individuals grouped into households

6.6.3 Improving the Data

Search out the highest quality source system

Examine the source to see how good/bad it is. Our favorite, and rather low-tech, approach is to perform a frequency count on each attribute to identify variations in spellings.

Correct variations in spelling, manually or preferably using a tool

Raise problems with Steering Committee

Fix problems at the source if at all possible

Fix some problems during data staging

If can't correct the data, be prepared to discuss where the data came from and why it looks the way it does

Use data cleansing tools against the data, and use trusted sources for correct values like address

Work with the source system owners to help them institute regular examination and cleansing of the source systems

Make the source system organizationally responsible for a clean extract

6.6.4 Data Quality Assurance

Define standards of acceptable data quality

Accuracy - documented audit trail that explains any differences between data warehouse and system of record

6.6.5 Basic Data Staging Audits

Correct number of rows processed

Referential integrity checking

Cross-footing

Set up series of queries against source system at different levels – compare to equivalent query against the data warehouse

Automate cross footing process

Manual examination

Look for numbers beyond acceptable ranges

Create set of 'reasonableness' data checks

Data staging process validation

Ensure that the process is sound

6.6.6 Data Access Audits

Check report logic / calculations

Confirm data access tool metadata set up properly

Review data content

Meaningless descriptions

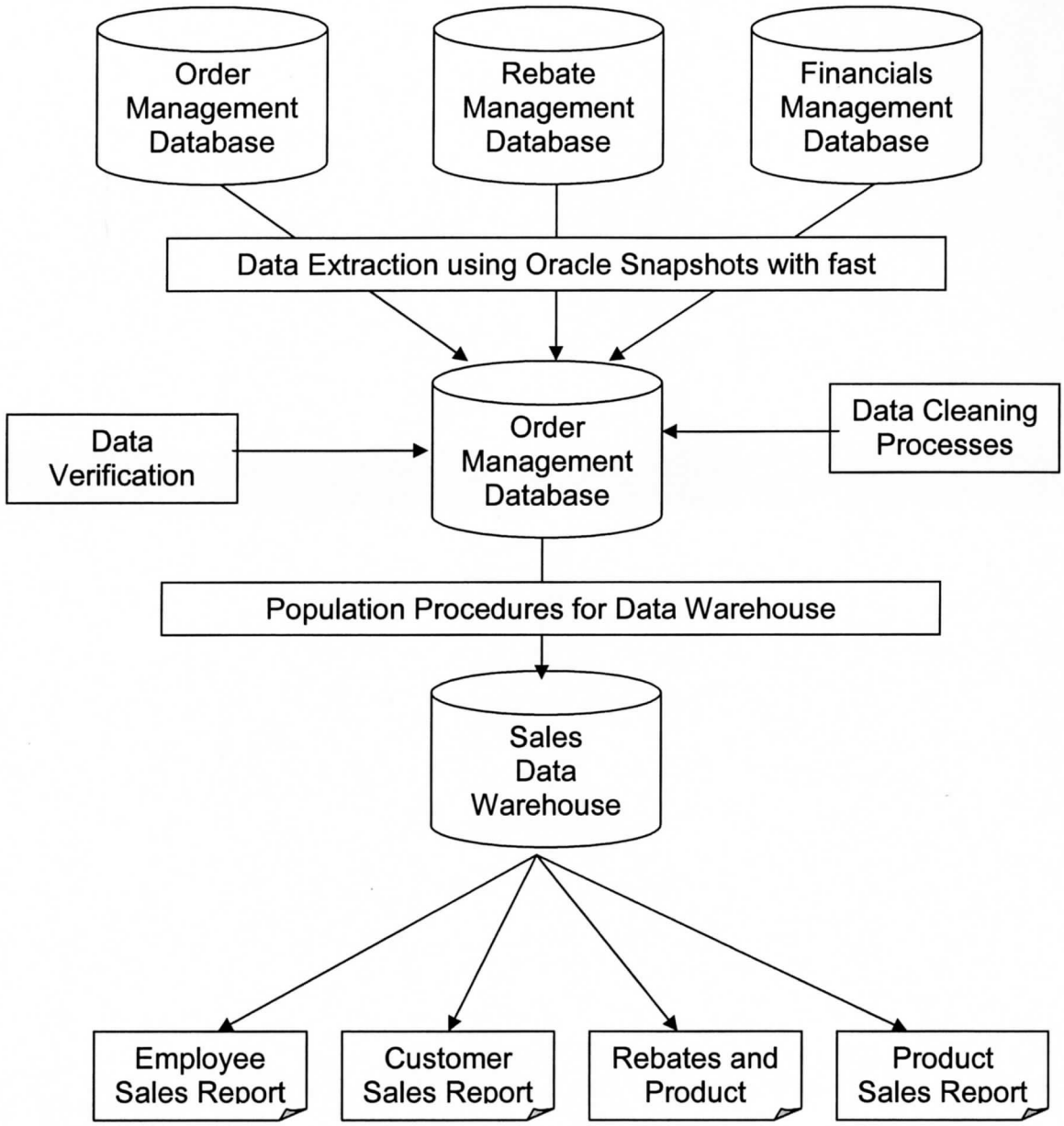
Duplicate dimension information

Incorrect dimensional relationships

Data not balancing

Data Staging Model

(Figure 5.7)



7. Implementation

Implementation is one of the very important tasks in data warehouse project. Since data warehouse is not a transactional system and its requirements for uptime, downtime and backup etc are completely different. So we do not need any specific method of implementation for a data warehouse.

Following are the steps we may need to perform in data warehouse implementation.

7.1 Installations

We need to install one test database on one of the available servers. As I mentioned before, we do have a mass storage devices in Solocup Data Center, so we may not be worrying about disk space. One of the RP 2405 HP servers is available for test. We need following software installed on the box: 1) Oracle 9.2.0.2 Enterprise Edition, HP-UX 11i with all the latest patch sets installed.

7.2 Database Creation

Create an Oracle database using OFA (Oracle Flexible Architecture) and with default tablespaces and data files. After database creation we need to create several new tablespace with a large size. The existing mass storage has been configured using RAID-5 and there are several RAID-5 sets are available. We need to ask Unix System Administrator to create file systems on different RAID-5 sets so that we could spread data and indexes on separate I/O channels. We need to consult Unix and Oracle Installation Manuals for details of installations.

7.3 Transforming Logical Data Model

Generate scripts to create tables and relationships among tables according to logical model. We developed several star schema logical models in design phase. For simplicity we showed Time Dimension and some other repeatedly in the start schemas, but we need to create only one table in the database. So we will create TIME DIMENTION, EMPLOYEE_DIMENSION, PRODUCT_DIMENTION, CUSTOMER_DIMENTION, DISTRICT_DIMENTION, REBATE_DIMENSION, EMPLOYEE_SALESFACT, CUSTOMER_SALESFACT, DISTRICT_SALESFACT, REBATE_SALESFACT, and PRODUCT_SALESFACT tables. The Primary Keys and Foreign Keys are mentioned in logical data model.

7.4 DW Population Procedures

7.4.1 The Extraction Process

The extraction process involves the movement of data from the source systems - in this case Order Management System, Rebate System and Financial System. This data could be in several different formats like flat files, export or a relational staging area. But Solocup has already very robust design of relational database systems and they are tightly integrated with each other using Oracle snapshots. Snapshots are Oracle provided methods of replicating data among the Oracle Databases. So I chose Oracle snapshots to extract data from all of above source systems, which will provide Solocup a better way of maintaining extraction process. Appendix G (Staging Model) has a general over view how and from which source systems data will be extracted.

All of the employee data is stored in the Financial System and it is propagated to Order Management and Rebate Systems through Oracle snapshots through out the day. Product and customer information is stored in Order Management System and it is propagated to Financial System and Rebate System through Oracle snapshots throughout the day. So I decided to extract data from the original base systems to the staging area using Oracle snapshots as per following.

7.4.2 The Loading Process

The loading phase moves the data from the staging area into the dimension and fact tables of the data warehouse proper. This is one of the most important parts of data warehouse population. Once data was extracted from source systems to a staging area the most important challenge was to clean data and to consolidate the data according to the dimensional requirements. Initially I decided to develop a PL/SQL, which we could run once in a night to populate data warehouse, but later I came to this conclusion that creating a trigger on the each snapshot would solve the problem, as Oracle is capable of pulling data from source systems only the changed or new data.

There are three source systems, Order Management System, Rebate System and Financial System, from which database warehouse will be populated. Following tables with desired columns are identified in each of above three systems to populate data warehouse.

1. **Order Management System:**

   PRODUCT (product_id, product_name, product_description, category, price)

   CUSTOMER (customer_id, name, type, address, city, state)

DISTRICT (district_id, name, description, manager, address, city, state)

SALES_DETAIL (customer_id, product_id, district_id, date, total_sales)

2. **Rebate System:**

   REBATE (rebate_id, type, description, city, state)

   REBATE_DETAIL (rebate_id, product_id, date, total_rebate)

3. **Financial System:**

   EMPLOYEE (employee_id, first_name, last_name, job, address, city,state)

Oracle snapshots will be used to bring data from above three systems into staging area where data will be cleaned, validated and added to the data warehouse. Following are the procedures, which are developed to clean, validate data in staging area and then adding to data warehouse.

7.4.3 Population Scripts

A complete set of scripts can be viewed in appendix I

7.5 Deployment Readiness Checklist

7.5.1 Desktop installation

Technology in place for business end user access to data warehouse
User logons and security authorizations obtained
Data Quality Verification And Reconciliation
Data quality assurance testing performed
Inconsistencies with historically reported investigated, resolved and documented
Beta team business representative signed-off on data legibility, completeness, and quality

7.5.2 End User Applications

End user applications developed and tested

Beta team business representative sign-off on application template quality and business relevance

7.5.3 End User Education

Introductory business end user education materials developed on data content, application templates, and end user tool usage

Beta team business representatives' sign-off on introductory user education offering

Education delivery logistics (e.g., venue, projection capabilities, user PCs with necessary data and application access, education materials duplication, etc.) handled

Production end users registered for education with appropriate approval from their managers

7.5.4 End User Support

Support organization in place and thoroughly trained

Support communication, bug report, and change request tracking procedures tested during beta period

7.6 Desktop Installation Readiness Checklist

Determine client configuration requirements to support end user data access software, including hardware configurations, ODBC connections, intranet and Internet connectivity, and so on.

Determine LAN addresses for the identified target users if you are not already using dynamically assigned LAN addresses.

Conduct a physical audit of the technology currently installed on these users' desks and compare it to the stated client configuration requirements.

Complete contract and procurement process to acquire any necessary client hardware, software, and/or upgrades.

Estimate lead-time required to acquire hardware, software and/or upgrades.

Acquire user logons and security approval as necessary for network and database access.

Estimate lead-time required to acquire user logins and security authorizations.

Ensure security maintenance procedures are in place (e.g., force changes in passwords with specified frequency).
Test installation procedures on a variety of machines. These procedures can be refined via the alpha and beta release processes discussed later in this chapter.
Schedule the installation with the users to align with their data warehouse education.
Install the hardware and/or software and complete installation testing. It is important to fully test each installation to verify the user's existing system has not been adversely impacted and to ensure that the appropriate connectivity have been established or retained.

7.7 User Accounts and Security Roles

We need to create several roles to access to users. Most of the roles will be read-only roles as there will be no updating, deletion or insertion by the users for any reason. However we can setup read-only roles according to type of access, disk and CPU utilization, importance and several other factors. Some of the batch jobs, which will be running reports for a long time, may not need high disk and CPU access times as compared to ad-hoc users.

7.8 Setup Maintenance Team

Data Warehouse needs proper maintenance on regular basis, so setting up a team to manage data warehouse is very important. Maintenance may include creating new indexes, backing up database, monitoring data warehouse population log files, tuning and other system management tasks. (See the Appendix J for tuning tips)

7.9 Tuning Tips

7.9.1 Tuning Goals

- Measurable (General)
- Response Time
- Database Availability
- Database Hit Percentage
- Memory Utilization
- Measurable (Quantitative)
- **Throughput**: Work per unit time as measured by transactions per seconds (TPS); higher the better.
- **Response Time:** Time it takes an Application to respond measured in seconds / milliseconds; lower the better.
- **Wall Time:** Elapsed Time a Program takes to run; lower the better.
- An **OLTP System** wants low Response Time and high Throughput.
- A **DSS System** wants low Response Time.
- A **Batch (Production) System** wants lower Wall Times.

7.9.2 Tuning tips

- Whenever possible to compensate for the electromechanical disadvantage (Memory being faster than Disk) Pre-Fetch. by using Caching, Buffering & Queuing .
- Access the least number of Oracle Data Blocks in Memory and Cache these in Memory.
- Read and Write Data as fast as possible. (May use Fast Disks, RAID and Parallel Operations)
- Share Application Code.
- Minimize CONTENTION
- Resources ...Users should not wait for Resources.
- Disk .....Use OS or Manual striping to speed up read and writes.

- Perform Backups and Housekeeping with minimizing impact.

7.9.3 Tuning Steps

- Tune the Design
- Do a Proper Logical Design.
- Do a Proper Physical Design.
- Redesign if Necessary.
- **Tune the Application**
- Write Efficient Application Code.
- Rewrite Code if necessary.
- **Tune Memory**
- Tune Database Memory Structures.
- Tune OS Memory Structures if necessary.
- **Tune I/O**
- Tune Database I/O.
- Tune OS I/O if necessary
- **Tune Contention**
- Disk
- Resources
- **Tune OS (As above)**
- **Tune Network** if necessary.
- **Tune Clients** if necessary.
- **IF all else fails consider more EXOTIC Solutions as under**:
- Multi Threaded Server (MTS). Maximizes throughput (Not Response Time).
- Transaction Processing (TP) Monitors.
- Parallel Query Option.
- Clustering
- Bitmapped Indexes (For DSS and OLAP Systems)
- Massively Parallel Processor Machines.

- Solid State Disks (Access Time is near to core memory speed).
- Hardware Accelerators
- Queuing Systems

7.9.4 Application Types

- OLTP ...Online Transaction Processing (e.g. Data Entry System)
- DSS.... Decision Support System (e.g. Database behind Intranet / Internet)
- HYBRID...(OLTP + DCSS)
- BATCH.... Production & Operation
- OLAP ...Online Analytical Processing ((e.g. GIS, Demographic Database)
- VCDB... Variable Cardinality Database (Short lived Security authorization DB)

7.9.5 Oracle Diagnostic Tools

- Dynamic Performance Views and Server manager Line Mode.
- Server Manager Monitor (GUI)
- The Performance pack of Enterprise Manager (OEM).
- Utlbstat / utlestat and report.txt
- Statspack
- SQL_TRACE, TKPROFF
- EXPLAIN PLAN
- AUTOTRACE
- Third Party Products (BMC, Platinum and Compuware).

7.9.6 Evaluating Performance

- **Is Hardware OK Is Hardware** including CPUs, Hard Disks, RAM, etc of both SERVER & CLIENTS compatible with the Oracle needs.

- **Is OS Tuned**
- **Check CPU Usage** (Not more than 90% Busy) 25-40% OS Processing & 60-75% Application Processing is desired. If CPU giving more time to OS Processing ; there is insufficient memory or Poor Application Design causing too many OS Calls.
- **Monitor Memory Usage** Detect and reduce Paging and swapping. Fix swap size in NT and UNIX and provide enough shared memory and semaphores in UNIX to give Oracle Processes enough Breathing Room to operate efficiently
- **Tune I/O** by balancing load across Disks and Controllers. Make use of many small instead of few large Disks.
- **Is Network Traffic** being efficiently handled by Network Hardware/ Software.
- **Is DB Tuned & OK**

- **OFA.** Oracle should be installed and configured using Oracle Optimal Flexible Architecture as it
  - Eliminates fragmentation
  - Minimize Contention
- **Avoid Dynamic Space Allocation**. Oracle stores data in storage units called segments. (Tables, Indexes, etc). Dynamic growth results in fragmentation thus hampering performance. Allocate enough space initially to avoid above.
- Data and Index Segments should be initialized properly.
- Temporary and Rollback Segments should be initialized properly.
- **Tuning Memory management** As Memory Access is faster than Disk Access, Oracle stores SQL and Data in Memory Structures.
- Evaluate and Fix the Memory Structures Size (for storing SQL Statements , associated Data and Transactions Log).
- Look for and reduce Paging
- All pages of SGA can be read into memory in one go if required. Even SGA can be locked into memory.
- **Reducing Disk I/O**

- By caching code and data.

- Match DBWRs with # of Disks remaining within the upper limit.
- By checking Row chaining and migration.
- Use Asynchronous I/O
- Identify and Fix Hot Files
- Identify and Fix excessive Disk and Tablespace Fragmentation.
- **Reducing Disk Contention**
- By OS Striping (RAID)
- By Manual Striping
- **Reducing Contention**
- Identify and Fix Resource Contention (Latches, Free List)
- Identify and Fix Rollback Segments Contention
- Identify and Fix Parallel Query Contention
- Identify and Fix Resource Contention for Parallel Server
- Avoid Index Contention
- Avoid Free List Contention
- Avoid Lock Contention
- **General**
- Conserving System Resources by creating and assigning Profiles.
- Ensuring Data Integrity through Integrity Constraints and Enforcing Custom Business Rules through Stored Procedures and riggers.
- Enforcing Data Security through Users Authentication, Privileges, Roles, etc

7.10 Sample Implementation

7.10.1 Table Creation Script (Appendix E)

7.10.2 Sample Data (Appendix F)

7.10.3 Sample Queries (Appendix G)

8. Conclusion

Solocup has several online systems on the latest hardware/software. The core systems are Order Management Database, Rebate Database and Financial Database. The online systems are being used as reporting tools and systems are being taxed heavily. This project was intended to demonstrate the use of the decision support tools and to separate the decision support data from online systems.

I completed the first three phases, planning; analysis and design, of the project within timeframe with the exception when interviews in the analysis phase took little longer than planned, but there were several issues in the implementation phase and this delayed the whole project. Following were the reasons for delaying of this phase:

- Hardware/Software resources were not available on time.
- There were several reports where appropriate data was not available. For example there was no aggregate data available on payments to brokers.
- Management took long time to approve budget for data warehousing projects and they hired data warehouse project manager by the end of May 2003.
- There are some serious conflicts within groups and departments, which were competing for resources that resulted in delay in implementation.

What I achieved in this project can be a good start to develop a full-fledge data warehouse. The design of star schema and ETL process is a good example to demonstrate the management about the flow and use of data in the data warehouse. Although several key areas like data integrality and validity processes were not

addressed in detail, the complexity and deployment of those processes became clear.

As things to do in future, I suggested management to setup a team of dedicated professional who could analyze in detail:

- All of the online systems
- Assess the report requirements
- Develop a robust ETL process
- Develop a comprehensive process for data integrity and validity
- Scaling data warehouse
- Develop and deploy user querying/reporting tools

References

1. Ralph Kimball, Laura Reeves, Margy Ross and Warren Thornthwaite, John Wiley & Sons (1998). The Data Warehouse Lifecycle Toolkit: Tools and Techniques for Designing, Developing, and Deploying Data Warehouses, 101-109, 189-190, and 240-259.

2. Ramez Elmasri & Shamkant B. Navathe, Addison-Wesley (2000). Fundamentals of Database Systems, Third Edition, 841-872.

3. William A. Giovinazzo, PH PTR (2000). Object-Oriented Data Warehouse Design (Building a Star Schema), 61-67, 158-176.

4. Robert Groth, PH PTR (2000). Data Mining (Building Comprehensive Advantage), 211-244.

5. Don Berleson, O.G.H 1998, High Performance Oracle Database Turning, 189-205.

6. Using RAID for performance and security, www.quest.com/whitepapers/Raid1.pdf

Appendix A (Interview Letter Template)

Dear ATTENDEE,

Thank you for participating in user meetings for the PROJECT NAME data warehouse project. As a reminder, the PROJECT NAME project is focused on … The objective of the user meetings is to better understand your area's business goals and priorities, which translate into data, and analyses needs. Your insight during these meetings is crucial to defining the requirements for PROJECT NAME. Specifically, project team members intend to discuss the following topics during their meeting with you:

Responsibilities

Business Objectives and Issues

Analyses and Data Requirements

Project Success Criteria

Please **bring copies of the analyses** you are currently performing and/or requesting.

ATTENDEE, thanks in advance for your participation. The project team looks forward to meeting you on DATE at TIME in MEETING ROOM. Please call me if you have any questions in the meantime.

Sincerely,

Executive Sponsor or Joint Project Managers

Appendix B (Interviews)

Interview with Susan Mark (Senior Voice President)

The following is the interview conducted with Susan Mark, senior voice president. Susan's primary role is to manage Information Services department and Sales Management systems. Susan started as project manager Sales and Marketing and after several years of being one of the top performers in his group she was promoted into middle level management. From there she worked her way up the corporate ladder to where she is today. Susan was first who backed the idea of Mr. Kearns to develop a Sales Data Warehouse with full support and resources.

Q: What do you see as the main objective of the data warehouse?

A: We had been experiencing loss of control on our system for past few years. Let me tell you, its Jim who convinced me about Data Warehouse. I fully back his ideas.

Q: What programs are you currently using to perform your analysis?

A: At the moment I use Excel and Word mainly. I also have some stuff the My Buddy put together. Reports and stuff like that.

Q: The My Buddy?

A: That's what I call Michael Mcpherren, my old buddy who runs the computer stuff for most of my requirements.

Q: Do you see this as something you will use personally?

A: Absolutely. That's the whole idea. I want that data in my hands as soon as it is available.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: Like I said, Excel and Word mainly. I have a new Dell. It is top of the line, black and everything.

Q: How would you describe yourself when it comes to computer usage?

A: I have been using one for years now.

Q: What parameters will determine the success of this project?

A: If it gives me what I want, when I need it AND it is finished on time and budget!!! I hear these things are always late and cost too much.

Q: Who currently enters the data and what data do they enter? Will this change with the new system? If yes how?

A: I don't know. I sure don't!

Q: How often do you update your data?

A: I like to look at it every day. Michael Mcpherren usually gives me reports in the morning and evening and sometimes I ask for special reports during the day.

Q: Do you want the information presented to you in graphs or as a summary?

A: Yes

Q: What kinds of questions do you want answered by the warehouse.

A: I want to be able to pull together data on all the customers and sales. I want to be able to see it at the client level, or for the whole company. I need to know stuff like sales overall, by item, by product. I want to be able to grab it anyway I want.

Q: Do you get some of this now?

A: Yes, Tom Sacher gives me reports with a lot of this stuff now.

Q: So why do you need the warehouse?

A: Cause the young man umm I mean the Jim Kearns wants it and thinks it will help us a lot in the long run.

Q: Do you do any trend analysis now?

A: Of Course!! That's my main job.

Q: So you don't really see the need for the warehouse?

A: Well, I wouldn't say that exactly. We'll see. It seams like a lot of time and money. If it can give me what I want then it will be worth it. I guess.

Q: Are there other things you would like to get from the data?

A: I would love to be able to get a handle on the inventory flow.

Q: Anything else?

A: Yeah. I need to get to know my markets better and the employees in the company, particularly sales people.

Q: Are you available for more questions if needed.

A: Yes sure!

Thanks Susan.

Interview with Jim Kearns (CIO)

Jim joined this company almost a year ago and since then he has worked really heard to make things better. In his first three months he gave company a vision that the company should be in next few years. He rebuilt a state of the art data center with the latest available technologies. He brought in some of the most energetic people in his team. A mass storage was setup to hold up all of the company data and a tape library to backup the entire data center in a central location. He also worked heard to make automate several routine maintenance tasks of the systems. Jim Kearns was first who realized to have a Sales Data Warehouse, which would play an important role in making strategic decisions.

Q: What do you see as the main objective of the data warehouse?

A: I have a big picture of Data Warehouse in my mind. I mean when I joined Solocup very soon I realized that people were losing control of the operations. Let me tell you, when the company was small, all the old concepts worked fine, but now in this business competitive environment I think information is everything and Data Warehouse is the only answer.

Q: What programs are you currently using to perform your analysis?

A: At the moment we are using a small database to keep sales and customers history. Excel and Word are also used to keep track of several sales and customer related data.

Q: How this database is populated?

A: Good question! I really don't know a lot of details yet, but one of my project managers told me that Oracle Snapshot have been used from our main online Sales/Customer database.

Q: Who is that project manager?

A: Tom Sacher. He is the one who is taking care of reporting from historical sales database.

Q: Is not that database is enough?

A: No, I think this is a poorly designed system, which is just a replica of original sales/customers data. I want a data warehouse, which should be designed and developed according to new technologies and software concepts.

Q: What do you mean by new technologies and software concepts?

A: I think I should ask you the same question.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: Like I said, Sales History Database, Excel and Word mainly. We are planning to move our entire data center to a centralized location. HP-UX and Oracle Database will be main software. All of our employees have desktops running windows/2000.

Q: How would you describe yourself when it comes to computer usage?

A: I have been using one for years now.

Q: What parameters will determine the success of this project?

A: If it gives me what I want, when I need it AND it is finished on time and budget!!! I hear these things are always late and cost too much.

Q: Who currently enters the data and what data do they enter? Will this change with the new system? If yes how?

A: I don't know. I sure don't!

Q: How often do you receive update your data requests?

A: I think every day. Valentina usually produces reports from sales history database and then she manages these reports in Excel sheets.

Q: Do you think people will like if the information presented to them in graphs or as a summary?

A: Yes

Q: What kinds of questions do you think people in the company are expecting to be answered by the warehouse?

A: I had several meetings with all the project managers in past several weeks. I think want detail data about a customer, a location, a product and item. Company runs promotions several times in a year and marketing people really need this data.

Q: Do you get some of this now?

A: Yes, but I'm not satisfied with the existing system and the ways reports are gathered.

Q: Do you think this data warehouse will fulfill all of their requirements?

A: Yes, I am fully convinced they will get more what they are getting now and with very less effort and resources.

Q: I'm glad Jim; you have that much positive view about data warehouse.

A: You are right. In my last company I had personally used a live data warehouse and used several data mining tools and the information presented was very effective.

Q: What were those data mining tools?

A: They were Oracle OLAP and Oracle Data Warehousing tools, I guess.

Q: Are you available for more questions if needed.

A: Kid, when I say its so, its so!! Call me anytime.

Thanks Jim.

Interview with Jim Lail (Senior Data Architect)

Jim Lail also joined this company recently. He is playing an important role in developing and implementing Middleware where all of the applications will communicate with each other. Jim has also very vast experience in database designs and developments. I think he will also play an important role in this project also.

Q: What do you see as the main objective of the data warehouse?

A: Well, I think data warehouse is the need for current business environments.

Q: What are the main systems where you were involved in Data Architectural work?

A: I really didn't. I'm mainly working on design and implementation of Middleware and our objective is to consolidate transactions running against several systems. Let me tell you, while doing this Middleware project I had to study a detailed data flow among several system and I think I know several things now

Q: Do you know anything about Sales History Database?

A: Yes, actually it is an exact replica of a few tables from Sales/Customer database, formally called Order Management. Data is replicated to Sales History database through Oracle Snapshots at the end of everyday. There is an in house developed

set of procedures in Oracle PL/SQL, which load data into Sales History database. This processes is taking almost 7-8 hours everyday.

Q: Do you think this Sales History database does not have which Data Warehouse will have?

A: Good question! I think Sales History database is not designed to store data date, time and dimensional wise. Data warehouse will have completely different set of tables according to need of data.

Q: Did you mean by Star Schema etc?

A: Exactly.

Q: How much we will be successful in building a Data Warehouse?

A: As long Jim Kearns is there, I think he will allocate as much resources as you will need. Also new Data Warehouse will have much more information than what the current Sales History Database has. Company has very good infrastructure for computing technologies and all the employees are well aware of computer use, so I think they will want it to be successful.

Q: Are you available for more questions if needed.

A: Anytime, just check my calendar and grab me for a meeting.

Thanks Jim.

Interview with Tom Sacher (Senior Project Manager)

Tom Sacher is working in Solocup for several years now. He joined the company as a sales assistant. He made his way up and he is a senior project manager of Musgistics and Sales History systems. Tom has a lot of experience in maintaining systems and he know the existing integration between different applications.

Q: What do you see as the main objective of the data warehouse?

A: Well, we already have one, which we are using for analysis purposes and my group produces several useful reports from that. I really don't know what Mr. Kearns wants when he says to build a new Data Warehouse.

Q: What programs are you currently using to perform your reporting for analysis?

A: At the moment we use Excel and Word mainly. We generate reports from Sales Data warehouse and then we bring them in Excel and Word.

Q: Do you think any problem of customizing reports in Excel?

A: No, we are doing this for several years.

Q: How you will feel if new Data Warehouse would provide you all the capabilities of reporting where you may not need Excel?

A: That is a very good idea and I like it as anybody else would, but I never saw such things to be successful. Anyhow I would say again, it would be great if we have one like that.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: I use Excel and Word mainly. I have a Dell Laptop and a Dell Desktop with windows/2000.

Q: How would you describe yourself when it comes to computer usage?

A: I'm using computer for last 10 years.

Q: What parameters will determine the success of this project?

A: If it gives me what I have to do now.

Q: Who currently enters the data and what data do they enter? Will this change with the new system? If yes how?

A: I don't know on online Sales and Customer database side, however I know data is replicated to Sales History database everyday in the evening.

Q: How often do you update your data in Excel?

A: Usually we run reports against Sales History database and then put in Excel where it is merged with existing reports. Sometime we get requests for report, which takes a lot of time of my staff as they have to group together information from Excel sheets. What a pain.

Q: Do people demand for graphical and summary reports too?

A: Yes

Q: Are you available for more questions if needed.

A: Yes sure!

Thanks Tom.

Interview with Roger Johnson (Manager Operations)

Roger is working in this company for almost 14 years now. He is running Operation Department as a supervisor for at least 6 years. He would be able tell about batch different processes running and also how long it takes to refresh Sales History database.

Q: Roger! Do you know we want to develop a Sales Data Warehouse, which will replace existing Sales History Database? What do you see as the main objective of the data warehouse?

A: I really don't know much about it, but all I know we have Sales History Database, which Tom and his group use for reporting purposes.

Q: Do you see any problem having a new database in your portfolio for operation?

A: That will matter at all. I think Mr. Kearns has a very big view for the company for over next few years and we will provide every support we can.

Q: Do you how Sales History database is refreshed?

A: We run a process daily between 5pm or 7pm, which takes about 7-8 hours to complete.

Q: What systems are involved?

A: Order Management database is the only one, which is used during this process.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: I use Excel and Word mainly. I have Dell Desktop with windows/2000.

Q: How would you describe yourself when it comes to computer usage?

A: I'm using computer for a long time now

Q: What parameters will determine the success of this project?

A: At the moment refresh of Sales History database takes a lot of time, which I think, should be reduced in new data warehouse database. I will prefer some fast refresh type of things if possible.

Q: Are you available for more questions if needed.

A: Yes sure!

Thanks Roger.

Interview with Kristen Meeker (Manager Order Management)

Kristen is working in Solocup Company for last 5 years. He joined as Manager Order Management and since then she is doing her job around managing orders. She is well familiar with several of Order Management process and I think she would contribute significant amount of knowledge sharing. She has been involved in analyzing Sales and Orders for a long time.

Q: What do you see as the main objective of the data warehouse?

A: I have been involved in managing Order Management for years now. I think for our current Sales History database is not so robust that we can rely on in the long run.

Q: What are the reasons you think we need a Sales Data Warehouse?

A: First of all, current Sales History database is no more than an exact replica of our online Order Management database. Some of the data is not stored time wise in Sales History database and it takes very long to get reports when joining tales together.

Q: What you would like to get more from Sales Data Warehouse?

A: I need to know the customer orders history along the final sales to a particular customer. Also Inventory is one thing that we do not have good control of that.

Q: Do you see this as something you will use personally?

A: Absolutely. That's the whole idea. I want that data in my hands as soon as it is available.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: Yes I do have my personal computer and I use mostly Excel and Word.

Q: How would you describe yourself when it comes to computer usage?

A: for about three years

Q: What parameters will determine the success of this project?

A: If it gives me what I want. I believe what Mr. Kearns is doing and I'm great fan of his future ideas.

Q: Who currently enters the data and what data do they enter? Will this change with the new system? If yes how?

A: We get orders from customers through several sources, like mail, email, and brokers and through our online web site. We have a group of people who enter orders in Order Management system and from there we track everything.

Q: How often do you update your data?

A: I like to look at it every day. Michael Mcpherren usually gives me reports in the morning and evening and sometimes I ask for special reports during the day also.

Q: Do you want the information presented to you in graphs or as a summary?

A: Yes

Q: Do you get some of this now?

A: Yes.

Q: What else Kristen, you would like from Sales Data Warehouse.

A: I would like sales by each employee and brokers.

Q: What is the function of brokers?

A: Well, they work just like our regular salesmen. They also work with some of the big clients like Wal-Mart and K-Mart to work promotions.

Q: If I need to talk to you any time later?

A: Sure, I'm available anytime to answer your questions. Oh, by the way just send me an email with any question. I usually check my emails regularly and I hope you will get my reply soon.

Thanks Kristen.

Interview with John Cheney (Manager Rebate Management)

John Cheney is around for more than eight years now. He joined the company as Assistant Credit Analyst. He is working as Manager rebate Management for last 3 years. John also keeps a close look at sales reports before and after a rebate is run. His role is very strategic and I think he would be a potential user of data warehouse.

Q: How Rebate is used for analysis?

A: Rebates are often run to attract new customers and to maintain existing customers. Whenever, we run a Rebate program, we analysis Sales over that period of time and also we compare sales with Rebate programs.

Q: How you get this information?

A: We get Sales report from Tom's group regularly and we keep on analysis the Rebate programs affects on sales.

Q: What do you know about the new Sales Data Warehouse project?

A: In our last meeting with Jim Kearns, he gave us a very good picture of data warehouse. I think one thing is very tempting and it is we will have direct access to Sales related information. At the moment we have to wait to get reports for hours and even days in some cases.

Q: Do you ask someone for special kind of reports?

A: Oh yes. I usually request Tom to get reports. When it comes to special reports, he takes very long time to get back as he has very limited resources and on of that they have to customize the report in Excel.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: Yes I do have my personal computer and I use mostly Excel and Word.

Q: How would you describe yourself when it comes to computer usage?

A: For several years now. I'm a computer science graduate by the way.

Q: What parameters will determine the success of this project?

A: If we get direct access to Sales Data Warehouse then I think we can get reports of our choice.

Q: How often do you update your data?

A: I like to look at it every day. Michael Mcpherren usually gives me reports in the morning and evening and sometimes I ask for special reports during the day also.

Q: Do you want the information presented to you in graphs or as a summary?

A: Yes

Q: Do you get some of this now?

A: Yes.

Q: Who else is involved when you run a rebate?

A: Brokers are the one source who works for us. They usually work with clients and they manage some of the stuff by their own.

Q: What kind of stuff they do?

A: They help clients to update their databases to print rebates and update their system etc.

Q: If I need to talk to you any time later?

A: Sure.

Thanks John.

Interview with Rick Harris (Warehouse Manager)

Rick is managing company warehouses for last two years. He is the one who keeps track of actual shipments to and from warehouse. He has a close look at the inventory and he sends strategic reports for inventory for head office once in a week. He may be a potential user of the warehouse too.

Q: What do you do?

A: I manage all the company warehouses at six different locations.

Q: What systems do you use?

A: We have one pallet management system, which we use to manager warehouses inventory.

Q: How it is related to Sales?

A: Well, this system is not directly related to sales except it keeps track of shipped orders. We send a report everyday to head office, which updates Order Management system.

Q: What is the format of that report?

A: We generate that report from pallet reporting system in a text format and it is loaded on Order Management using Sql-Loader.

Q: Are the pallet systems directly connected with each other?

A: No, they are on just stand-alone computers.

Q: How would you describe yourself when it comes to computer usage?

A: I'm using computer for years now.

Q: Do you see Sales report?

A: Well!!! No.

Q: Do you want the information presented to you in graphs or as a summary?

A: Yes

Q: Do you get some of this now?

A: Yes.

Q: If I need to talk to you any time later?

A: Sure.

Thanks Rick.

Interview with Dan Leuchinsiky (Senior Sales Manager)

I found Dan as the actual user of Data Warehouse, who routinely requests Sales reports from Tom's group. Dan keeps a track of reports on his desktop too and he reports Sales trend to higher management and he is extensively involved in making strategic decisions. Dan would be the main user of data warehouse and his input would be very help to analyze data warehouse requirements.

Q: What do you see as the main objective of the data warehouse?

A: Oh boy!!! I have been working in this company for last seven years and this is first time I think we got a CIO who is taking some of very bold initiatives. In his last meeting with project managers, I fully supported his idea of building Sales Data Warehouse. We have a state of the art data center, but I think we are under utilizing it. We are still organizing our reports in Excel and also we do not have much control too.

Q: What programs are you currently using to perform your analysis?

A: At the moment I use Excel and Word mainly. I get Excel sheet from Tom's group and then I modify myself sometime.

Q: Do you have a personal computer or workstation? If so, what programs do you use?

A: Like I said, Excel and Word mainly. I have a new Dell, which we recently bought for several company employees.

Q: How would you describe yourself when it comes to computer usage?

A: Oh boy!!! I am using one for several years now.

Q: What parameters will determine the success of this project?

A: I am mostly interested in Sales, Markets, Brokers and customer analysis. I want a trend at each location and client. You know we have several big clients like Wal-Mart, Kmart and McDonald's etc. We also run rebates over the period of time and after that we always need to analysis of sales and feed back from customers.

Q: Who currently enters the data and what data do they enter? Will this change with the new system? If yes how?

A: I don't know for sure.

Q: How often do you update your data?

A: I like to look at it every day. I keep on requesting Tom's group for several reports which I think they get from Sales History Database.

Q: Do you want the information presented to you in graphs or as a summary?

A: That's the whole idea, according to Mr. Kearns, we will have data in Sales Data Warehouse distributed by dimensions and there are several tools in the market, which can put that data into graphs.

Q: What kinds of questions do you want answered by the warehouse.

A: As I said earlier, I would like to see Sales by customer, by market, by employees, by location and by type. I want to be able to pull together data on all the customers and sales. I want to be able to see it at the client level, or for the whole company. I need to know stuff like sales overall, by item, by product. I want to be able to grab it anyway I want.

Q: Do you get some of this now?

A: Yes, but Tom's group is very busy and it takes us long time to get some strategic reports.

Q: So why do you need the warehouse?

A: You know, one thing that I liked in our last meeting Mr. Kearns, we will have a direct access to Sales Data Warehouse and they will provide us with our own logins and access tools and we will be able to get data any way we would like. That will awesome, if that happens.

Q: Do you do any trend analysis now?

A: Of Course!! That's my main job.

Q: Are there other things you would like to get from the data?

A: I would love to be able to get a handle on the inventory flow and warehouse status.

Q: Anything else?

A: Yeah. I need to get to know my markets better and the employees in the company, particularly sales people.

Q: I think I had a very good discussion with you Dan! I hope I may need to talk to you again. Could we meat again next week and I'm really interested in reports which you manage by your own and that ones which you get from Tom's group. And also I would like your ideas how you would like reports from the new Data Warehouse

A: Yes sure! Just make sure you arrange a meeting with me on the calendar.

Thanks Dan.

Interview with Michael McPherren (Senior Developer)

Mike is a senior developer and he has been involved in developing several in-house software projects including existing Sales History database refresh packages and procedures. His input would be greatly helpful to understand the optimal refresh methods of data warehouse.

Q: Do you know about new Sales Data Warehouse Project?

A: Yes, I heard from Jim Kearns, that he has given a new vision for IS department in next 2 years and developing a data warehouse is one of the key project in his plans.

Q: Mike, how much you were involved in developing Sales History Database refresh processes?

A: It's almost 2 years now when I wrote several PI/SQL Packages, which extract data from Order Management Database using snapshots, and then they refresh Sales History Database.

Q: How much you are involved in developing reports from Sales History Database?

A: Well, I often get calls from Susan Mark about some of the customized reports from Sales History Database. Dan also calls me for almost same kind of reports as Susan.

Q: What do you think Sales History Database does not provide?

A: Current Sales History database is almost an exact replica of Order Management database. It is so much normalized and while getting reports it takes very long time. Also when querying database which involves data dimensional wise, it's almost very ugly.

Q: What parameters will determine the success of this project?

A: I think if we provide a well-designed database to users, we will get a very good feedback from users. I am convinced with proper implementation of start schemas it will not be much difficult to store and extract database, which is required, by most of the reports.

Q: How often do you get request fro reports?

A: Very often.

Q: Do you get request where user want the information presented in graphs or as a summary?

A: Yes, that is why we mostly bring data in Excel and from there we print graphs etc.

Q: What kinds of questions do you want answered by the warehouse.

A: As I said earlier, Dan and Susan like to see Sales by customer, by market, by employees, by location and by type. They want to be able to pull together data on all the customers and sales. They also want to be able to see it at the client level, or for the whole company. They need to know stuff like sales overall, by item, by product.

Q: How much experience you have in Data Warehouse?

A: Well, I did not work on Data Warehouse project so far, but I'm very good in writing PL/SQL code, so when it comes this I'm here.

Q: I think I had a very good discussion with you Mike! I hope I may need to talk to you again. Could we meat again next week?

A: Yes sure!

Thanks Mike.

Appendix C (DW Readiness Litmus Test)

Data Warehouse Readiness Litmus Test

Factor	Low Readiness	⇔	High Readiness
Strong Business Management Sponsor			
	Not well respected	⇔	Considerable organizational clout
	Can take weeks for team to gain access	⇔	Readily available to team
	"I'll get back to you on that"	⇔	Quick, decisive resolution to issues
	Hope "you" get it done	⇔	Active, vocal and visible supporter – willing to put own neck on line
	You can deliver this to 250 users next month, right?	⇔	Realistic expectations
	"A data whatta?"	⇔	Data warehouse savvy
Compelling Business Motivation			
	"And your point is?"	⇔	Survival dependent on data warehouse
	Funding is a big problem	⇔	Cost is not an issue – we can't afford not to do this!
	"Shifting sands" vision	⇔	Clearly articulated vision
	Ten different views of the solution	⇔	Consistent view of the solution
	Tactical issue	⇔	Strategic issue
	Cost savings opportunity	⇔	Incremental revenue opportunity
	Unable to quantify the payback	⇔	Huge payback
IS / Business Partnership			
	Business engages outside consultant without IS knowledge	⇔	Business and IS work hand-in-hand
	Business unit creates own pseudo IS team to	⇔	IS actively engaged with business unit

	build a data warehouse		
	"We can't trust any numbers from our systems"	⇔	Strong confidence in existing reporting environment
	It takes "years" to get a new ad hoc request turned around	⇔	Quick IS response to ad hoc requests
	Users don't even submit requests anymore	⇔	Short existing user request backlog

Current Analytic Culture			
	"Gut feel" decision making	⇔	Decision making relies on facts and figures
	Users don't ask for data	⇔	Business users clamor for access to data – "Just get me the data and I'll figure it out"
	Users don't look at current reports	⇔	Current reports are consistently rekeyed into spreadsheets for analysis and historical trending
	Current reports used as doorstops until the recycling bin comes by	⇔	Current reports are dog eared, highlighted and filled with yellow self adhesive notes
	Users have secretaries log on and print off email to read it	⇔	Users are very computer literate
	Finance is extremely possessive of bottom line performance figures	⇔	Information shared openly throughout the organization
Feasibility			
	Data warehouse would require purchase of all new technology	⇔	Robust technical infrastructure in place
	Everyone and their uncle is committed to the year 2000 project	⇔	Experienced resources available

	Reliable data won't be available until after the enterprise resource planning (ERP) implementation	⇔	Quality data available

Appendix D (User Acceptance Form Template)

<Deliverable Name> - User Acceptance

Project: _____

Executive Sponsor: _____

Business Driver: _____

Project Manager _____

Deliverable Description _____ **:**

Accepted:

_____	_____
Approval Signature	Project Manager Signature
_____	_____
Approval Name	Project Manager Name
_____	_____
Date	Date

Appendix E (Tables Creation Script)

```
DROP TABLE CUSTOMER_DIM CASCADE CONSTRAINTS ;

CREATE TABLE CUSTOMER_DIM (
 CUSTOMER_ID  NUMBER       NOT NULL,
 CUST_NAME    VARCHAR2 (30),
 CUST_TYPE    VARCHAR2 (10),
 ADDRESS      VARCHAR2 (30),
 CITY         VARCHAR2 (20),
 STATE        VARCHAR2 (2))
  TABLESPACE SYSTEM
  PCTFREE 10
  PCTUSED 40
  INITRANS 1
  MAXTRANS 255
 STORAGE (
  INITIAL 65536
  MINEXTENTS 1
  MAXEXTENTS 2147483645
  FREELISTS 1 FREELIST GROUPS 1 )
  NOCACHE;


ALTER TABLE CUSTOMER_DIM
 ADD CONSTRAINT PK_CUSTOMER_DIM
  PRIMARY KEY (CUSTOMER_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE CUSTOMER_SALESFACT CASCADE CONSTRAINTS ;

CREATE TABLE CUSTOMER_SALESFACT (
 CUSTOMER_ID  NUMBER       NOT NULL,
 TIME_ID      NUMBER       NOT NULL,
 PRODUCT_ID   NUMBER       NOT NULL,
 DOLLARSALES  NUMBER)
  TABLESPACE SYSTEM
  PCTFREE 10
  PCTUSED 40
  INITRANS 1
  MAXTRANS 255
 STORAGE (
  INITIAL 65536
```

```
  MINEXTENTS 1
  MAXEXTENTS 2147483645
  FREELISTS 1 FREELIST GROUPS 1 )
  NOCACHE;


ALTER TABLE CUSTOMER_SALESFACT
 ADD CONSTRAINT PK_CUSTOMER_SALESFACT
  PRIMARY KEY (CUSTOMER_ID, TIME_ID, PRODUCT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE DISTRICT_DIM CASCADE CONSTRAINTS ;

CREATE TABLE DISTRICT_DIM (
 DISTRICT_ID  NUMBER       NOT NULL,
 NAME        VARCHAR2 (20),
 DESCRIPTION  VARCHAR2 (50),
 MANAGER      NUMBER,
 ADDRESS      VARCHAR2 (30),
 CITY        VARCHAR2 (20),
 STATE       VARCHAR2 (2))
  TABLESPACE SYSTEM
  PCTFREE 10
  PCTUSED 40
  INITRANS 1
  MAXTRANS 255
  STORAGE (
  INITIAL 65536
  MINEXTENTS 1
  MAXEXTENTS 2147483645
  FREELISTS 1 FREELIST GROUPS 1 )
  NOCACHE;


ALTER TABLE DISTRICT_DIM
 ADD CONSTRAINT PK_DISTRICT
  PRIMARY KEY (DISTRICT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE DISTRICT_SALESFACT CASCADE CONSTRAINTS ;
```

```
CREATE TABLE DISTRICT_SALESFACT (
 DISTRICT_ID  NUMBER       NOT NULL,
 TIME_ID      NUMBER       NOT NULL,
 PRODUCT_ID   NUMBER       NOT NULL,
 DOLLARSALES  NUMBER)
  TABLESPACE SYSTEM
  PCTFREE 10
  PCTUSED 40
  INITRANS 1
  MAXTRANS 255
 STORAGE (
  INITIAL 65536
  MINEXTENTS 1
  MAXEXTENTS 2147483645
  FREELISTS 1 FREELIST GROUPS 1 )
  NOCACHE;


ALTER TABLE DISTRICT_SALESFACT
 ADD CONSTRAINT PK_DISTRICT_SALESFACT
  PRIMARY KEY (DISTRICT_ID, TIME_ID, PRODUCT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE EMPLOYEE_DIM CASCADE CONSTRAINTS ;

CREATE TABLE EMPLOYEE_DIM (
 EMPLOYEE_ID  NUMBER       NOT NULL,
 FIRST_NAME   VARCHAR2 (20),
 LAST_NAME    VARCHAR2 (20),
 CITY         VARCHAR2 (20),
 STATE        VARCHAR2 (2),
 JOB_TITLE    VARCHAR2 (20),
 MANAGER      NUMBER)
  TABLESPACE SYSTEM
  PCTFREE 10
  PCTUSED 40
  INITRANS 1
  MAXTRANS 255
 STORAGE (
  INITIAL 65536
  MINEXTENTS 1
  MAXEXTENTS 2147483645
```

```
  FREELISTS 1 FREELIST GROUPS 1 )
  NOCACHE;


ALTER TABLE EMPLOYEE_DIM
 ADD CONSTRAINT PK_EMPLOYEE_DIM
  PRIMARY KEY (EMPLOYEE_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE EMPLOYEE_SALESFACT CASCADE CONSTRAINTS ;

CREATE TABLE EMPLOYEE_SALESFACT (
  EMPLOYEE_ID  NUMBER       NOT NULL,
  TIME_ID      NUMBER       NOT NULL,
  PRODUCT_ID   NUMBER       NOT NULL,
  DOLLARSALES  NUMBER)
   TABLESPACE SYSTEM
   PCTFREE 10
   PCTUSED 40
   INITRANS 1
   MAXTRANS 255
  STORAGE (
   INITIAL 65536
   MINEXTENTS 1
   MAXEXTENTS 2147483645
   FREELISTS 1 FREELIST GROUPS 1 )
   NOCACHE;


ALTER TABLE EMPLOYEE_SALESFACT
 ADD CONSTRAINT PK_EMPLOYEE_SALESFACT
  PRIMARY KEY (EMPLOYEE_ID, TIME_ID, PRODUCT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE PRODUCT_DIM CASCADE CONSTRAINTS ;

CREATE TABLE PRODUCT_DIM (
  PRODUCT_ID          NUMBER       NOT NULL,
  PRODUCT_NAME        VARCHAR2 (20),
  PRODUCT_DESCRIPTION  VARCHAR2 (50),
```

```
  CATAGORY          VARCHAR2 (10))
   TABLESPACE SYSTEM
   PCTFREE 10
   PCTUSED 40
   INITRANS 1
   MAXTRANS 255
  STORAGE (
   INITIAL 65536
   MINEXTENTS 1
   MAXEXTENTS 2147483645
   FREELISTS 1 FREELIST GROUPS 1 )
   NOCACHE;


ALTER TABLE PRODUCT_DIM
 ADD CONSTRAINT PK_PRODUCT_DIM
  PRIMARY KEY (PRODUCT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE PRODUCT_SALESFACT CASCADE CONSTRAINTS ;

CREATE TABLE PRODUCT_SALESFACT (
  TIME_ID     NUMBER       NOT NULL,
  PRODUCT_ID  NUMBER       NOT NULL,
  DOLLARSALES  NUMBER)
   TABLESPACE SYSTEM
   PCTFREE 10
   PCTUSED 40
   INITRANS 1
   MAXTRANS 255
  STORAGE (
   INITIAL 65536
   MINEXTENTS 1
   MAXEXTENTS 2147483645
   FREELISTS 1 FREELIST GROUPS 1 )
   NOCACHE;


ALTER TABLE PRODUCT_SALESFACT
 ADD CONSTRAINT PK_PRODUCT_SALESFACT
  PRIMARY KEY (TIME_ID, PRODUCT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
```

```
;


DROP TABLE REBATE_DIM CASCADE CONSTRAINTS ;

CREATE TABLE REBATE_DIM (
  REBATE_ID    NUMBER       NOT NULL,
  REBATE_TYPE  VARCHAR2 (20),
  DESCRIPTION  VARCHAR2 (30))
   TABLESPACE SYSTEM
   PCTFREE 10
   PCTUSED 40
   INITRANS 1
   MAXTRANS 255
  STORAGE (
   INITIAL 65536
   MINEXTENTS 1
   MAXEXTENTS 2147483645
   FREELISTS 1 FREELIST GROUPS 1 )
   NOCACHE;


ALTER TABLE REBATE_DIM
 ADD CONSTRAINT PK_REBATE
  PRIMARY KEY (REBATE_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE REBATE_SALESFACT CASCADE CONSTRAINTS ;

CREATE TABLE REBATE_SALESFACT (
  REBATE_ID    NUMBER       NOT NULL,
  TIME_ID      NUMBER       NOT NULL,
  PRODUCT_ID   NUMBER       NOT NULL,
  DOLLARREBATE  NUMBER,
  DOLLARSALES   NUMBER)
   TABLESPACE SYSTEM
   PCTFREE 10
   PCTUSED 40
   INITRANS 1
   MAXTRANS 255
  STORAGE (
   INITIAL 65536
   MINEXTENTS 1
```

```
  MAXEXTENTS 2147483645
  FREELISTS 1 FREELIST GROUPS 1 )
  NOCACHE;


ALTER TABLE REBATE_SALESFACT
 ADD CONSTRAINT PK_REBATE_SALESFACT
  PRIMARY KEY (REBATE_ID, TIME_ID, PRODUCT_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


DROP TABLE TIME_DIM CASCADE CONSTRAINTS ;

CREATE TABLE TIME_DIM (
  TIME_ID  NUMBER       NOT NULL,
  YEAR     NUMBER,
  MONTH    NUMBER,
  WEEK     NUMBER,
  DAY      NUMBER)
   TABLESPACE SYSTEM
   PCTFREE 10
   PCTUSED 40
   INITRANS 1
   MAXTRANS 255
  STORAGE (
   INITIAL 65536
   MINEXTENTS 1
   MAXEXTENTS 2147483645
   FREELISTS 1 FREELIST GROUPS 1 )
   NOCACHE;


ALTER TABLE TIME_DIM
 ADD CONSTRAINT PK_TIME_DIM
  PRIMARY KEY (TIME_ID)
  USING INDEX
  TABLESPACE SYSTEM PCTFREE 10  STORAGE(INITIAL 65536 )
;


ALTER TABLE CUSTOMER_SALESFACT ADD  CONSTRAINT FK_CUSTOMER
 FOREIGN KEY (CUSTOMER_ID)
  REFERENCES CUSTOMER_DIM (CUSTOMER_ID) ;
```

```
ALTER TABLE CUSTOMER_SALESFACT ADD  CONSTRAINT
FK_CUSTOMER_PRODUCT
 FOREIGN KEY (PRODUCT_ID)
  REFERENCES PRODUCT_DIM (PRODUCT_ID) ;

ALTER TABLE CUSTOMER_SALESFACT ADD  CONSTRAINT
FK_CUSTOMER_TIME
 FOREIGN KEY (TIME_ID)
  REFERENCES TIME_DIM (TIME_ID) ;

ALTER TABLE DISTRICT_DIM ADD  CONSTRAINT FK_EMPLOYEE_DIS
 FOREIGN KEY (MANAGER)
  REFERENCES EMPLOYEE_DIM (EMPLOYEE_ID) ;

ALTER TABLE DISTRICT_SALESFACT ADD  CONSTRAINT FK_DISTRICT
 FOREIGN KEY (DISTRICT_ID)
  REFERENCES DISTRICT_DIM (DISTRICT_ID) ;

ALTER TABLE DISTRICT_SALESFACT ADD  CONSTRAINT
FK_DISTRICT_PRODUCT
 FOREIGN KEY (PRODUCT_ID)
  REFERENCES PRODUCT_DIM (PRODUCT_ID) ;

ALTER TABLE DISTRICT_SALESFACT ADD  CONSTRAINT FK_DISTRICT_TIME
 FOREIGN KEY (TIME_ID)
  REFERENCES TIME_DIM (TIME_ID) ;

ALTER TABLE EMPLOYEE_SALESFACT ADD  CONSTRAINT FK_EMPLOYEE
 FOREIGN KEY (EMPLOYEE_ID)
  REFERENCES EMPLOYEE_DIM (EMPLOYEE_ID) ;

ALTER TABLE EMPLOYEE_SALESFACT ADD  CONSTRAINT
FK_EMPLOYEE_PRODUCT
 FOREIGN KEY (PRODUCT_ID)
  REFERENCES PRODUCT_DIM (PRODUCT_ID) ;

ALTER TABLE EMPLOYEE_SALESFACT ADD  CONSTRAINT
FK_EMPLOYEE_TIME
 FOREIGN KEY (TIME_ID)
  REFERENCES TIME_DIM (TIME_ID) ;

ALTER TABLE PRODUCT_SALESFACT ADD  CONSTRAINT FK_PRODUCT
 FOREIGN KEY (PRODUCT_ID)
  REFERENCES PRODUCT_DIM (PRODUCT_ID) ;
```

```
ALTER TABLE PRODUCT_SALESFACT ADD  CONSTRAINT
FK_PRODUCT_TIME
 FOREIGN KEY (TIME_ID)
  REFERENCES TIME_DIM (TIME_ID) ;

ALTER TABLE REBATE_SALESFACT ADD  CONSTRAINT FK_REBATE
 FOREIGN KEY (REBATE_ID)
  REFERENCES REBATE_DIM (REBATE_ID) ;

ALTER TABLE REBATE_SALESFACT ADD  CONSTRAINT
FK_REBATE_PRODUCT
 FOREIGN KEY (PRODUCT_ID)
  REFERENCES PRODUCT_DIM (PRODUCT_ID) ;

ALTER TABLE REBATE_SALESFACT ADD  CONSTRAINT FK_REBATE_TIME
 FOREIGN KEY (TIME_ID)
  REFERENCES TIME_DIM (TIME_ID) ;

CREATE SEQUENCE CUST_SEQ INCREMENT BY 1 START WITH 1 NOCYCLE
CACHE 20 NOORDER;
CREATE SEQUENCE DIST_SEQ INCREMENT BY 1 START WITH 1 NOCYCLE
CACHE 20 NOORDER;
CREATE SEQUENCE EMP_SEQ INCREMENT BY 1 START WITH 1 NOCYCLE
CACHE 20 NOORDER;
CREATE SEQUENCE PROD_SEQ INCREMENT BY 1 START WITH 1 NOCYCLE
CACHE 20 NOORDER;
CREATE SEQUENCE REB_SEQ INCREMENT BY 1 START WITH 1 NOCYCLE
CACHE 20 NOORDER;
CREATE SEQUENCE TIME_SEQ INCREMENT BY 1 START WITH 1 NOCYCLE
CACHE 20 NOORDER;
```

Appendix F (Sample Data)

```
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
2, 2000, 12, 1, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
3, 2000, 12, 2, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
4, 2000, 12, 3, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
5, 2000, 12, 4, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
6, 2001, 12, 1, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
7, 2001, 12, 2, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
8, 2001, 12, 3, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
9, 2001, 12, 4, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
10, 2002, 12, 1, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
11, 2002, 12, 2, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
12, 2002, 12, 3, NULL);
INSERT INTO TIME_DIM ( TIME_ID, YEAR, MONTH, WEEK, DAY ) VALUES (
13, 2002, 12, 4, NULL);
COMMIT;

INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
1, 'Jim', 'Kulaga', 'Highland Park', 'IL', 'Manager', NULL);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
2, 'Kin', 'Mclaphen', 'Highland Park', 'IL', 'Manager', NULL);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
3, 'Mike', 'Mcpherren', 'Highland Park', 'IL', 'Manager', NULL);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
4, 'John', 'William', 'Urbana', 'IL', 'Salesman', 1);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
```

```
MANAGER ) VALUES (
5, 'Roger', 'Johnson', 'Urbana', 'IL', 'Salesman', 1);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
6, 'Kim', 'Kulaga', 'Twin Falls', 'MD', 'Salesman', 2);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
7, 'Irshad', 'Khan', 'Collumbus', 'OH', 'Salesman', 2);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
8, 'Steve', 'Martin', 'Rochester', 'NY', 'Salesman', 3);
INSERT INTO EMPLOYEE_DIM ( EMPLOYEE_ID, FIRST_NAME, LAST_NAME,
CITY, STATE, JOB_TITLE,
MANAGER ) VALUES (
9, 'Harry', 'Hareesha', 'Atlanta', 'GA', 'Salesman', 3);
COMMIT;


INSERT INTO PRODUCT_DIM ( PRODUCT_ID, PRODUCT_NAME,
PRODUCT_DESCRIPTION,
CATAGORY ) VALUES (
1, 'Plates 12', '12 inch Platic Plates', 'Plain');
INSERT INTO PRODUCT_DIM ( PRODUCT_ID, PRODUCT_NAME,
PRODUCT_DESCRIPTION,
CATAGORY ) VALUES (
2, 'Plates 14', '14 inch Platic Plates', 'Plain');
INSERT INTO PRODUCT_DIM ( PRODUCT_ID, PRODUCT_NAME,
PRODUCT_DESCRIPTION,
CATAGORY ) VALUES (
3, 'Bowls 12OZ', '12OZ Bowls', 'Deep');
INSERT INTO PRODUCT_DIM ( PRODUCT_ID, PRODUCT_NAME,
PRODUCT_DESCRIPTION,
CATAGORY ) VALUES (
4, 'Lid 16', '16 inch Lid', 'Plain');
INSERT INTO PRODUCT_DIM ( PRODUCT_ID, PRODUCT_NAME,
PRODUCT_DESCRIPTION,
CATAGORY ) VALUES (
5, 'Lid 12', '12 inch Lid', 'Plain');
INSERT INTO PRODUCT_DIM ( PRODUCT_ID, PRODUCT_NAME,
PRODUCT_DESCRIPTION,
CATAGORY ) VALUES (
6, 'Lid 10', '10 inch Lid', 'Coffee');
COMMIT;
```

```
INSERT INTO CUSTOMER_DIM ( CUSTOMER_ID, CUST_NAME, CUST_TYPE,
ADDRESS, CITY,
STATE ) VALUES (
1, 'Wal Mart', 'Retail', '1600 Ohio Dr', 'Chicago', 'IL');
INSERT INTO CUSTOMER_DIM ( CUSTOMER_ID, CUST_NAME, CUST_TYPE,
ADDRESS, CITY,
STATE ) VALUES (
2, 'Burger King', 'Food', '21 Oakland Ave', 'Gurnee', 'IL');
INSERT INTO CUSTOMER_DIM ( CUSTOMER_ID, CUST_NAME, CUST_TYPE,
ADDRESS, CITY,
STATE ) VALUES (
3, 'Mcdonold', 'Food', '108 City View Dr', 'Rochester', 'NY');
INSERT INTO CUSTOMER_DIM ( CUSTOMER_ID, CUST_NAME, CUST_TYPE,
ADDRESS, CITY,
STATE ) VALUES (
4, 'Jewle', 'Grocery', '21 Gurnee Ave', 'Los. A.', 'CA');
INSERT INTO CUSTOMER_DIM ( CUSTOMER_ID, CUST_NAME, CUST_TYPE,
ADDRESS, CITY,
STATE ) VALUES (
5, 'Costco', 'Retail', '871 Atlantic Blvd.', 'Jacksonville', 'FL');
COMMIT;

INSERT INTO DISTRICT_DIM ( DISTRICT_ID, NAME, DESCRIPTION, MANAGER,
ADDRESS, CITY,
STATE ) VALUES (
1, 'East Caost', 'District Sales Off for Eastern Regoin', NULL, NULL, NULL, NULL);
INSERT INTO DISTRICT_DIM ( DISTRICT_ID, NAME, DESCRIPTION, MANAGER,
ADDRESS, CITY,
STATE ) VALUES (
2, 'West Caost', 'District Sales Off for Western Regoin', NULL, NULL, NULL, NULL);
INSERT INTO DISTRICT_DIM ( DISTRICT_ID, NAME, DESCRIPTION, MANAGER,
ADDRESS, CITY,
STATE ) VALUES (
3, 'Midwest', 'District Sales Off for midwest Regoin', NULL, NULL, NULL, NULL);
INSERT INTO DISTRICT_DIM ( DISTRICT_ID, NAME, DESCRIPTION, MANAGER,
ADDRESS, CITY,
STATE ) VALUES (
4, 'South Coast', 'District Sales Off for southern Regoin', NULL, NULL, NULL,
NULL);
INSERT INTO DISTRICT_DIM ( DISTRICT_ID, NAME, DESCRIPTION, MANAGER,
ADDRESS, CITY,
STATE ) VALUES (
5, 'Europe', 'District Sales Off for europeon Regoin', NULL, NULL, NULL, NULL);
COMMIT;
```

```
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 3, 2, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 4, 3, 1000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 5, 4, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 6, 1, 1000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 7, 2, 4000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 8, 3, 3000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 9, 4, 800);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 10, 1, 1000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 11, 2, 1000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 12, 3, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 13, 4, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
```

```
DOLLARSALES ) VALUES (
6, 2, 1, 42000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 2, 1, 22000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 3, 2, 23000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 4, 3, 10000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 5, 4, 25000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 6, 1, 15000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 7, 2, 41000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 8, 3, 31000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 9, 4, 8100);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 10, 1, 11000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 11, 2, 18000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 12, 3, 28000);
```

```
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 13, 4, 20000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 2, 1, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 3, 2, 52000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 4, 3, 71000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 5, 4, 22000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 6, 1, 71000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 7, 2, 24000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 8, 3, 73000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 9, 4, 8800);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 10, 1, 41000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 11, 2, 71000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
```

```
DOLLARSALES ) VALUES (
6, 12, 3, 12000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
6, 13, 4, 92000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 2, 1, 4000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 3, 2, 5000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 4, 3, 7000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 5, 4, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 6, 1, 7000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 7, 2, 2000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 8, 3, 7000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 9, 4, 800);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 10, 1, 4000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 11, 2, 7000);
```

```
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 12, 3, 1000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
7, 13, 4, 9000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 2, 1, 84000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 3, 2, 75000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 4, 3, 67000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 5, 4, 52000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 6, 1, 47000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 7, 2, 32000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 8, 3, 27000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 9, 4, 9823);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 10, 1, 64000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
```

```
DOLLARSALES ) VALUES (
8, 11, 2, 77000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 12, 3, 81000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
8, 13, 4, 19000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 2, 1, 83000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 3, 2, 71000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 4, 3, 61000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 5, 4, 59000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 6, 1, 44000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 7, 2, 36000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 8, 3, 28000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 9, 4, 9123);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 10, 1, 69000);
```

```
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 11, 2, 71000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 12, 3, 84000);
INSERT INTO EMPLOYEE_SALESFACT ( EMPLOYEE_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
9, 13, 4, 49000);
COMMIT;

INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 3, 2, 2000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 4, 3, 1000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 5, 4, 2000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 6, 1, 1000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 7, 2, 4000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 8, 3, 3000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 9, 4, 800);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 10, 1, 1000);
```

```
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 11, 2, 1000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 12, 3, 2000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 13, 4, 2000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 2, 1, 42000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 2, 1, 22000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 3, 2, 23000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 4, 3, 10000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 5, 4, 25000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 6, 1, 15000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 7, 2, 41000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 8, 3, 31000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
```

```
DOLLARSALES ) VALUES (
2, 9, 4, 8100);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 10, 1, 11000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 11, 2, 18000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 12, 3, 28000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 13, 4, 20000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 6, 1, 71000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 7, 2, 24000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 8, 3, 73000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 9, 4, 8800);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 10, 1, 41000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 11, 2, 71000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 12, 3, 12000);
```

```
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 13, 4, 92000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 2, 1, 4000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 3, 2, 5000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 4, 3, 7000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 5, 4, 2000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 6, 1, 7000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 7, 2, 2000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 8, 3, 7000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 9, 4, 800);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 10, 1, 4000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 11, 2, 7000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
```

```
DOLLARSALES ) VALUES (
4, 12, 3, 1000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 13, 4, 9000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 2, 1, 84000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 3, 2, 75000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 4, 3, 67000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 5, 4, 52000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 6, 1, 47000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 7, 2, 32000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 8, 3, 27000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 9, 4, 9823);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 10, 1, 64000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 11, 2, 77000);
```

```
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 12, 3, 81000);
INSERT INTO CUSTOMER_SALESFACT ( CUSTOMER_ID, TIME_ID,
PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 13, 4, 19000);
COMMIT;

INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 3, 2, 552000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 4, 3, 661000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 5, 4, 222000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 6, 1, 991000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 7, 2, 884000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 8, 3, 663000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 9, 4, 77800);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 10, 1, 221000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 11, 2, 91000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 12, 3, 662000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 13, 4, 72000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
1, 2, 1, 342000);
```

```
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 2, 1, 922000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 3, 2, 323000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 4, 3, 710000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 5, 4, 725000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 6, 1, 315000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 7, 2, 841000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 8, 3, 431000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 9, 4, 98100);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 10, 1, 811000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 11, 2, 718000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 12, 3, 728000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
2, 13, 4, 620000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 6, 1, 671000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 7, 2, 424000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 8, 3, 673000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
```

```
DOLLARSALES ) VALUES (
3, 9, 4, 88800);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 10, 1, 241000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 11, 2, 771000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 12, 3, 912000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
3, 13, 4, 992000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 2, 1, 24000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 3, 2, 75000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 4, 3, 67000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 5, 4, 52000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 6, 1, 67000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 7, 2, 52000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 8, 3, 47000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 9, 4, 67800);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 10, 1, 54000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 11, 2, 97000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
```

```
4, 12, 3, 91000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
4, 13, 4, 59000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 2, 1, 684000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 3, 2, 375000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 4, 3, 667000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 5, 4, 552000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 6, 1, 747000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 7, 2, 432000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 8, 3, 927000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 9, 4, 999823);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 10, 1, 864000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 11, 2, 587000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 12, 3, 491000);
INSERT INTO DISTRICT_SALESFACT ( DISTRICT_ID, TIME_ID, PRODUCT_ID,
DOLLARSALES ) VALUES (
5, 13, 4, 239000);
COMMIT;
```

Appendix G (Tables with data and Sample Queries)

TIME_DIM

TIME_ID	YEAR	MONTH	WEEK	DAY
2	2000	12	1	
3	2000	12	2	
4	2000	12	3	
5	2000	12	4	
6	2001	12	1	
7	2001	12	2	
8	2001	12	3	
9	2001	12	4	
10	2002	12	1	
11	2002	12	2	
12	2002	12	3	
13	2002	12	4	

EMPLOYEE_DIM

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	CITY	STATE	JOB_TITLE	MANAGER
1	Jim	Kulaga	Highland Park	IL	Manager	
2	Kin	Mclaphen	Highland Park	IL	Manager	
3	Mike	Mcpherren	Highland Park	IL	Manager	
4	John	William	Urbana	IL	Salesman	1
5	Roger	Johnson	Urbana	IL	Salesman	1
6	Kim	Kulaga	Twin Falls	MD	Salesman	2
7	Irshad	Khan	Collumbus	OH	Salesman	2
8	Steve	Martin	Rochester	NY	Salesman	3
9	Harry	Hareesha	Atlanta	GA	Salesman	3

PRODUCT_DIM

PRODUCT_ID	PRODUCT_NAME	PRODUCT_DESCRIPTION	CATAGORY
1	Plates 12	12 inch Platic Plates	Plain
2	Plates 14	14 inch Platic Plates	Plain
3	Bowls 12OZ	12OZ Bowls	Deep
4	Lid 16	16 inch Lid	Plain
5	Lid 12	12 inch Lid	Plain
6	Lid 10	10 inch Lid	Coffee

CUSTOMER_DIM

CUSTOMER_ID	CUST_NAME	CUST_TYPE	ADDRESS	CITY	STATE
1	Wal Mart	Retail	1600 Ohio Dr	Chicago	IL
2	Burger King	Food	21 Oakland Ave	Gurnee	IL

3	**Mcdonold**	**Food**	**108 City View Dr**	**Rochester**	**NY**
4	**Jewle**	**Grocery**	**21 Gurnee Ave**	**Los. A.**	**CA**
5	**Costco**	**Retail**	**871 Atlantic Blvd.**	**Jacksonville**	**FL**

DISTRICT_DIM

DISTRICT_ID	NAME	DESCRIPTION	MANAGER	ADDRESS	CITY	STATE
1	East Caost	District Sales Off for Eastern Regoin				
2	West Caost	District Sales Off for Western Regoin				
3	Midwest	District Sales Off for midwest Regoin				
4	South Coast	District Sales Off for southern Regoin				
5	Europe	District Sales Off for europeon Regoin				

EMPLOYEE_SALESFACT

EMPLOYEE_ID	TIME_ID	PRODUCT_ID	DOLLARSALES
5	3	2	2000
5	4	3	1000
5	5	4	2000
5	6	1	1000
5	7	2	4000
5	8	3	3000
5	9	4	800
5	10	1	1000
5	11	2	1000
5	12	3	2000
5	13	4	2000
6	2	1	42000
4	2	1	22000
4	3	2	23000
4	4	3	10000
4	5	4	25000
4	6	1	15000
4	7	2	41000
4	8	3	31000
4	9	4	8100
4	10	1	11000
4	11	2	18000
4	12	3	28000

4	13	4	20000
5	2	1	2000
6	3	2	52000
6	4	3	71000
6	5	4	22000
6	6	1	71000
6	7	2	24000
6	8	3	73000
6	9	4	8800
6	10	1	41000
6	11	2	71000
6	12	3	12000
6	13	4	92000
7	2	1	4000
7	3	2	5000
7	4	3	7000
7	5	4	2000
7	6	1	7000
7	7	2	2000
7	8	3	7000
7	9	4	800
7	10	1	4000
7	11	2	7000
7	12	3	1000
7	13	4	9000
8	2	1	84000
8	3	2	75000
8	4	3	67000
8	5	4	52000
8	6	1	47000
8	7	2	32000
8	8	3	27000
8	9	4	9823
8	10	1	64000
8	11	2	77000
8	12	3	81000
8	13	4	19000
9	2	1	83000
9	3	2	71000
9	4	3	61000
9	5	4	59000
9	6	1	44000
9	7	2	36000
9	8	3	28000
9	9	4	9123
9	10	1	69000
9	11	2	71000
9	12	3	84000
9	13	4	49000

CUSTOMER_SALES_FACT

CUSTOMER_ID	TIME_ID	PRODUCT_ID	DOLLARSALES
1	3	2	2000

1	4	3	1000
1	5	4	2000
1	6	1	1000
1	7	2	4000
1	8	3	3000
1	9	4	800
1	10	1	1000
1	11	2	1000
1	12	3	2000
1	13	4	2000
1	2	1	42000
2	2	1	22000
2	3	2	23000
2	4	3	10000
2	5	4	25000
2	6	1	15000
2	7	2	41000
2	8	3	31000
2	9	4	8100
2	10	1	11000
2	11	2	18000
2	12	3	28000
2	13	4	20000
3	6	1	71000
3	7	2	24000
3	8	3	73000
3	9	4	8800
3	10	1	41000
3	11	2	71000
3	12	3	12000
3	13	4	92000
4	2	1	4000
4	3	2	5000
4	4	3	7000
4	5	4	2000
4	6	1	7000
4	7	2	2000
4	8	3	7000
4	9	4	800
4	10	1	4000
4	11	2	7000
4	12	3	1000
4	13	4	9000
5	2	1	84000
5	3	2	75000
5	4	3	67000
5	5	4	52000
5	6	1	47000
5	7	2	32000
5	8	3	27000
5	9	4	9823
5	10	1	64000
5	11	2	77000

5	12	3	81000
5	13	4	19000

DISTRICT-SALESFACT

DISTRICT_ID	TIME_ID	PRODUCT_ID	DOLLARSALES
1	3	2	552000
1	4	3	661000
1	5	4	222000
1	6	1	991000
1	7	2	884000
1	8	3	663000
1	9	4	77800
1	10	1	221000
1	11	2	91000
1	12	3	662000
1	13	4	72000
1	2	1	342000
2	2	1	922000
2	3	2	323000
2	4	3	710000
2	5	4	725000
2	6	1	315000
2	7	2	841000
2	8	3	431000
2	9	4	98100
2	10	1	811000
2	11	2	718000
2	12	3	728000
2	13	4	620000
3	6	1	671000
3	7	2	424000
3	8	3	673000
3	9	4	88800
3	10	1	241000
3	11	2	771000
3	12	3	912000
3	13	4	992000
4	2	1	24000
4	3	2	75000
4	4	3	67000
4	5	4	52000
4	6	1	67000
4	7	2	52000
4	8	3	47000
4	9	4	67800
4	10	1	54000
4	11	2	97000
4	12	3	91000
4	13	4	59000
5	2	1	684000
5	3	2	375000
5	4	3	667000
5	5	4	552000

5	6	1	747000
5	7	2	432000
5	8	3	927000
5	9	4	999823
5	10	1	864000
5	11	2	587000
5	12	3	491000
5	13	4	239000

Query to get yearly sales by each employees

```
Select c.first_name,c.last_name,b.year,sum(a.dollarsales) dollarsales
from employee_salesfact a,time_dim b, employee_dim c, product_dim d
where a.employee_id=c.employee_id
and a.time_id=b.time_id
and a.product_id=d.product_id
Group by c.first_name,c.last_name,b.year
```

Result

FIRST_NAME	LAST_NAME	YEAR	DOLLARSALES
Kim	Kulaga	2000	187000
Kim	Kulaga	2001	176800
Kim	Kulaga	2002	216000
John	William	2000	80000
John	William	2001	95100
John	William	2002	77000
Harry	Hareesha	2000	274000
Harry	Hareesha	2001	117123
Harry	Hareesha	2002	273000
Roger	Johnson	2000	7000
Roger	Johnson	2001	8800
Roger	Johnson	2002	6000
Steve	Martin	2000	278000
Steve	Martin	2001	115823
Steve	Martin	2002	241000
Irshad	Khan	2000	18000
Irshad	Khan	2001	16800
Irshad	Khan	2002	21000

Query to get annual sales to each customer

```
Select c.cust_name,c.cust_type,b.year,sum(a.dollarsales) dollarsales
from customer_salesfact a,time_dim b, customer_dim c, product_dim d
where a.customer_id=c.customer_id
and a.time_id=b.time_id
and a.product_id=d.product_id
Group by c.cust_name,c.cust_type,b.year
```

CUST_NAME	CUST_TYPE	YEAR	DOLLARSALES
Jewle	Grocery	2000	18000
Jewle	Grocery	2001	16800
Jewle	Grocery	2002	21000
Costco	Retail	2000	278000
Costco	Retail	2001	115823
Costco	Retail	2002	241000
Mcdonold	Food	2001	176800
Mcdonold	Food	2002	216000
Wal Mart	Retail	2000	47000
Wal Mart	Retail	2001	8800
Wal Mart	Retail	2002	6000
Burger King	Food	2000	80000
Burger King	Food	2001	95100
Burger King	Food	2002	77000

Query to get annula sales report for all sales district order by last year and sales

```
Select c.name,c.description,b.year,sum(a.dollarsales) dollarsales
from district_salesfact a,time_dim b, district_dim c, product_dim d
where a.district_id=c.district_id
and a.time_id=b.time_id
and a.product_id=d.product_id
Group by c.name,c.description,b.year
Order by 3 desc ,4 desc
```

Result

NAME	DESCRIPTION	YEAR	DOLLARSALES
Midwest	District Sales Off for midwest Regoin	2002	2916000
West Caost	District Sales Off for Western Regoin	2002	2877000
Europe	District Sales Off for europeon Regoin	2002	2181000
East Caost	District Sales Off for Eastern Regoin	2002	1046000
South Coast	District Sales Off for southern Regoin	2002	301000
Europe	District Sales Off for europeon Regoin	2001	3105823
East Caost	District Sales Off for Eastern Regoin	2001	2615800
Midwest	District Sales Off for midwest Regoin	2001	1856800
West Caost	District Sales Off for Western Regoin	2001	1685100
South Coast	District Sales Off for southern Regoin	2001	233800
West Caost	District Sales Off for Western Regoin	2000	2680000
Europe	District Sales Off for europeon Regoin	2000	2278000
East Caost	District Sales Off for Eastern Regoin	2000	1777000
South Coast	District Sales Off for southern Regoin	2000	218000

Appendix H (Work Breakdown Structure)

Work Breakdown Structure – Solocup Sales Data Warehouse

PLANNING

- Complete Project Idea Form
- Introduction of Project
- Statement of Problem
- Evaluating Project Objective
- Defining Project Scope
  - Planning
  - Analysis
  - Design
  - Implementation
- Establish Project Timeline
- Work Breakdown Structure
- Gantt Chart
- Form a Graduate Committee
  - Dr. Ronghua Shan
  - Dr. Terry Dennis
  - Prof. Stephen
  - Sartaj H. Iram
- Prepare Project Planning Report
- Submit Project Planning Report for Approval

ANALYSIS

- Prepare and Plan for Interviews
- Conduct Interviews
  - Susan Mark (Senior Voice President)
  - Jim Kearns (CIO)
  - Jim Lail (Senior Data Architect)
  - Tom Sacher (Senior Project Manager)
  - Roger Johnson (Operation Manager)

- Krinsten Meeker (Manager Order Management)
- John Cheney (Manager Rebate Management)
- Rick Harris (Warehouse Manager)
- Dan Leuchinsiky (Senior Sales Manager)
- Michael Mcpherren (Developer)

Study Existing Databases and Applications

- Order Management Systems
- Warehouse Management Systems
- Rebates Management Systems

Determine Dataware Software Applications

Determine Hardware Requirements

Determine Database Requirements

- Oracle RDBMS
- SQL-Server

Build Analysis Model

Build Database Object Model

DESIGN

Design Logical Data Model

Design Relational Tables

Design Star Schemas

Design Indexes

Binary Indexes

Bitmap Indexes

Design Data Warehouse Population Methods

- Using flat files
- Using export of data
- Using database triggers

Design Data Staging Model

Design data cleaning methods

- Data Verification
- Data Quality Control

- Design Reports

IMPLEMENTATION

- Populate Dataware House from Staging area
- Setup user accounts and roles
- Perform Initial Tuning
- Allow users to test and print reports
- Setup a team for maintaining DW
    - Data Extraction Management
    - Data Staging Management
    - Adding more reports
    - Modifying data according to user requirements

Appendix I (Population Scripts)

```
CREATE SNAPSHOT PRODUCT
NOLOGGING
TABLESPACE USERS
STORAGE
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select  product_id, product_name, product_description, category, price from
PRODUCT @ORDMGMT.SNAP;

CREATE OR REPLACE TRIGGER PRODUCT_TRIG
ON PRODUCT AFTER INSERT FOR EACH ROW
BEGIN
INSERT INTO PRODUCT_DIM
(PRODUCT_ID, PRODUCT_NAME, PRODUCT_DESCRIPTION, CATEGORY,
PRICE)
VALUES
(:NEW.PRODUCT_ID, :NEW.PRODUCT_NAME,
:NEW.PRODUCT_DESCRIPTION, :NEW.CATEGORY, :NEW.PRICE)


CREATE SNAPSHOT CUSTOMER
NOLOGGING
TABLESPACE USERS
STORAGE
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select customer_id, name, type, address, city, state from
CUSTOMER@ORDMGMT.SNAP;

CREATE OR REPLACE TRIGGER CUSTOMER_TRIG
ON CUSTOMER AFTER INSERT FOR EACH ROW
BEGIN
INSERT INTO CUSTOMER_DIM
(CUSTOMER_ID, NAME, TYPE, ADDRESS, CITY, STATE, ZIP)
```

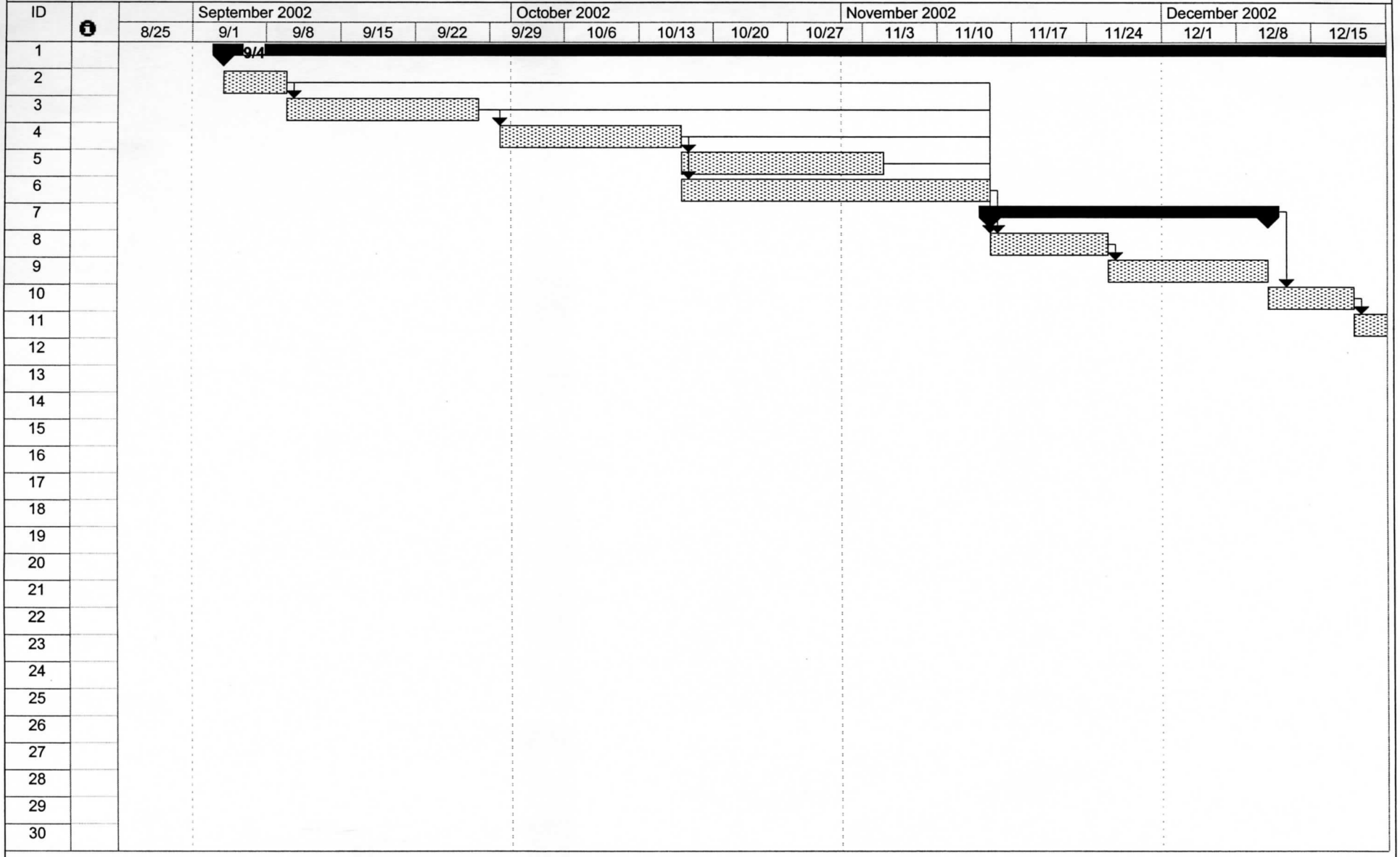
```
VALUES
(:NEW.CUSTOMER_ID, :NEW.NAME, :NEW.TYPE, :NEW.ADDRESS, :NEW.CITY,
:NEW.STATE, :NEW.ZIP)

CREATE SNAPSHOT DISTRICT
NOLOGGING
TABLESPACE USERS
STORAGE
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select district_id, name, description, manager, address, city, state from
DISTRICT@ORDMGMT.SNAP);

CREATE OR REPLACE TRIGGER DISTRICT_TRIG
ON DISTRICT AFTER INSERT FOR EACH ROW
BEGIN
INSERT INTO DISTRICT_DIM
(DISTRICT_ID, NAME, DESCRIPTION, MANAGER, ADDRESS, CITY, STATE)
VALUES
(:NEW.DISTRICT_ID, :NEW.NAME, :NEW.DESCRIPTION, :NEW.MANAGER,
:NEW.ADDRESS, :NEW.CITY, :NEW.STATE)

CREATE SNAPSHOT SALES_DETAIL
NOLOGGING
TABLESPACE USERS
STORAGE
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select customer_id, product_id, district_id, date, total_sales from
SALES_DETAIL@ORDMGMT.SNAP);

CREATE SNAPSHOT REBATE
NOLOGGING
TABLESPACE USERS
STORAGE
```

```
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select rebate_id, type, description, city, state from REBATE@ORDMGMT.SNAP);

CREATE OR REPLACE TRIGGER REBATE_TRIG
ON REBATE AFTER INSERT FOR EACH ROW
BEGIN
INSERT INTO REBATE_DIM
(REBATE_ID, TYPE, DESCRIPTION, CITY, STATE)
VALUES
(:NEW.REBATE_ID, :NEW.TYPE, :NEW.DESCRIPTION, :NEW.CITY,
:NEW.STATE)


CREATE SNAPSHOT REBATE_DETAIL
NOLOGGING
TABLESPACE USERS
STORAGE
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select rebate_id, product_id, date, total_rebate from
REBATE_DETAIL@ORDMGMT.SNAP);

CREATE SNAPSHOT EMPLOYEE
NOLOGGING
TABLESPACE USERS
STORAGE
(INITIAL 250K NEXT 32K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
USING INDEX TABLESPACE INDX
STORAGE
(INITIAL 32K NEXT 16K MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0)
REFRESH FORCE START WITH SYSDATE NEXT null FOR UPDATE AS
select employee_id, first_name, last_name, job, address, city,state from
EMPLOYEE@ORDMGMT.SNAP);
```

```
CREATE OR REPLACE TRIGGER EMPLOYEE_TRIG
ON EMPLOYEE AFTER INSERT FOR EACH ROW
BEGIN
INSERT INTO EMPLOYEE_DIM
(EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB, ADDRESS, CITY, STATE)
VALUES
(:NEW.EMPLOYEE_ID, :NEW.FIRST_NAME, :NEW.LAST_NAME, :NEW.JOB,
:NEW.ADDRESS, :NEW.CITY, :NEW.STATE)


CREATE PROCEDURE UPDATE_FACT_TABS
as
BEGIN
INSERT INTO EMPLOYEE_SALESFACT
SELECT EMPLOYEE_ID, PRODUCT_ID, get_timeid (date) "TIME_ID"
SUM(TOTAL_SALES)
FROM SALES_DETAIL
GROUP BY EMPLOYEE_ID, PRODUCT_ID, TIME_ID;

INSERT INTO CUSTOMER_SALESFACT
SELECT CUSTOMER_ID, PRODUCT_ID, get_timeid (date) "TIME_ID"
SUM(TOTAL_SALES)
FROM SALES_DETAIL
GROUP BY CUSTOMER_ID, PRODUCT_ID, TIME_ID;

INSERT INTO DISTRICT_SALESFACT
SELECT DISTRICT_ID, PRODUCT_ID, get_timeid (date) "TIME_ID"
SUM(TOTAL_SALES)
FROM SALES_DETAIL
GROUP BY DISTRICT_ID, PRODUCT_ID, TIME_ID;

INSERT INTO PRODUCT_SALESFACT
SELECT PRODUCT_ID, get_timeid (date) "TIME_ID" SUM(TOTAL_SALES)
FROM SALES_DETAIL
GROUP BY PRODUCT_ID, TIME_ID;

INSERT INTO REBATE_SALESFACT
SELECT REBATE_ID, PRODUCT_ID, get_timeid (date) "TIME_ID"
SUM(TOTAL_REBATE)
FROM REBATE_DETAIL
GROUP BY REBATE_IDPRODUCT_ID, TIME_ID;

DELETE FROM SALESFACT;

DELETE FROM REBATE_SALESFACT;
```

```
COMMIT;

END;

CREATE FUNCTION GET_TIMEID (vDATE)
As
MTime_ID number;
BEGIN
SELECT time_id into mTIME_ID FROM TIME_DIM
WHERE YEAR=to_char(vdate,'YYYY')
AND MONTH= to_char(vdate,'MON')
AND WEEK= to_char(vdate,'WEEK')
AND DAY=to_char(vdate,'DAY');

RETURN(mTIME_ID)

END;
```

Appendix J (Gantt Chart)



Project: Gantt Chart
Date: Wed 10/1/03

Task	Milestone	External Tasks
Split	Summary	External Milestone
Progress	Project Summary	Deadline

ID		September 2002					October 2002					November 2002				December 2002		
		8/25	9/1	9/8	9/15	9/22	9/29	10/6	10/13	10/20	10/27	11/3	11/10	11/17	11/24	12/1	12/8	12/15
31																		
32																		
33																		
34																		
35																		
36																		
37																		
38																		
39																		
40																		
41																		
42																		
43																		
44																		
45																		
46																		
47																		
48																		
49																		
50																		
51																		
52																		
53																		
54																		
55																		
56																		
57																		
58																		
59																		
60																		

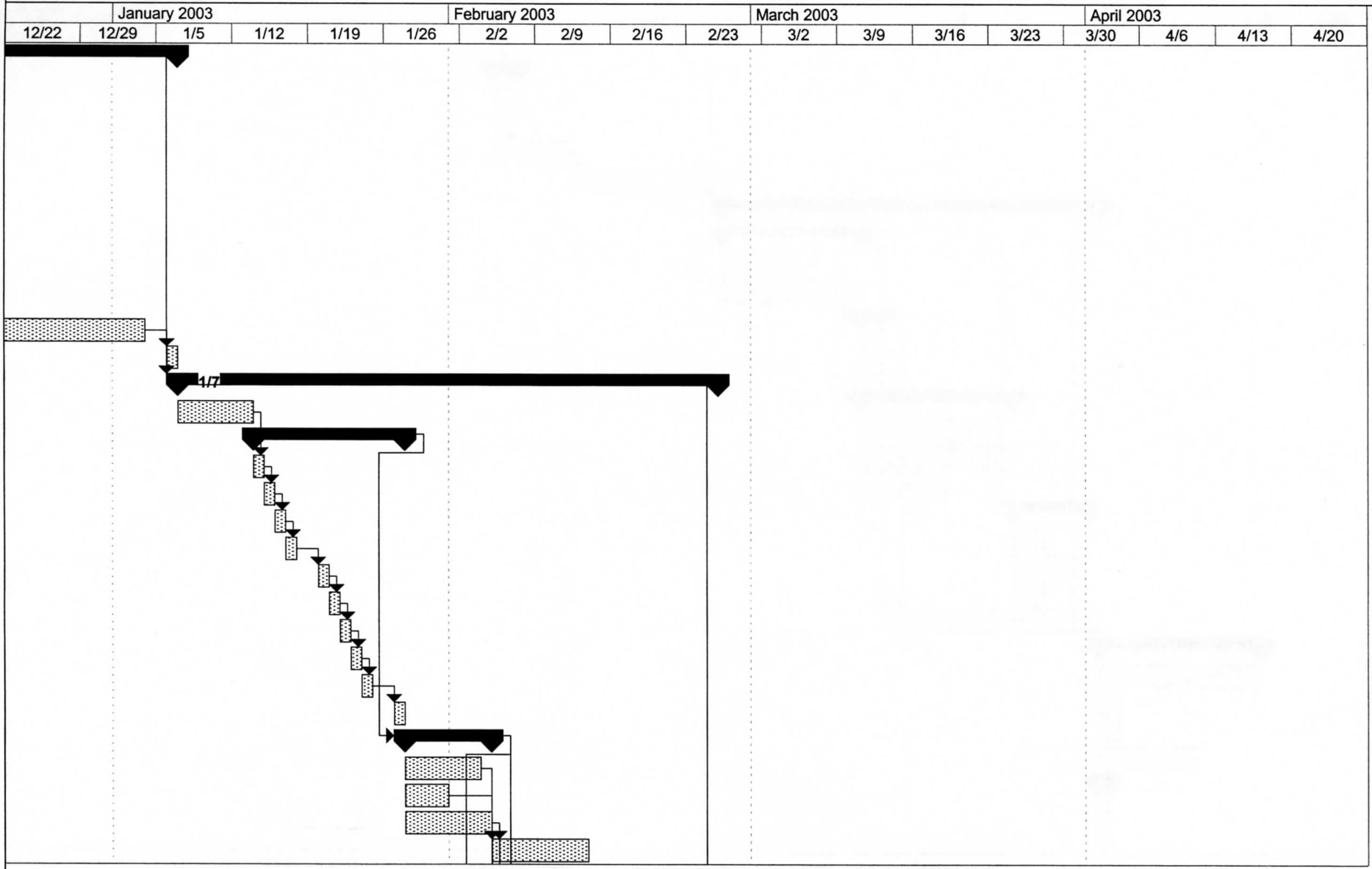
Project: Gantt Chart
Date: Wed 10/1/03

Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline

ID			September 2002				October 2002					November 2002				December 2002		
		8/25	9/1	9/8	9/15	9/22	9/29	10/6	10/13	10/20	10/27	11/3	11/10	11/17	11/24	12/1	12/8	12/15
61																		
62																		

Project: Gantt Chart
Date: Wed 10/1/03

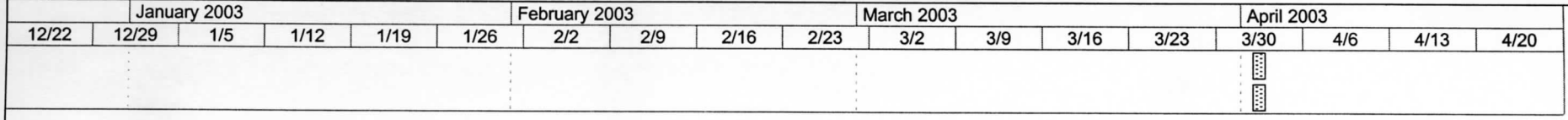
Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	



Project: Gantt Chart
Date: Wed 10/1/03

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	

Appendix J (Gantt Chart)



Project: Gantt Chart
Date: Wed 10/1/03

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	